NATIONAL INSTRUMENTS
LabVIEW
SignalExpress

LabVIEW™ SignalExpress™ Help

June 2008, 371268J-01

LabVIEW SignalExpress is an interactive measurement program you can use to quickly acquire, analyze, and present data from hundreds of data acquisition devices and instruments, with no programming required.

Use LabVIEW SignalExpress to create projects that acquire, analyze, create, generate, and display signals. You can acquire signals from hardware devices installed on your computer, analyze the signals, and then send the resulting signals to hardware devices.

The steps you use to create projects depend on the LabVIEW SignalExpress assistants you have installed. Refer to the documentation for an assistant to learn more about creating projects with assistant-specific steps.

You can use LabVIEW SignalExpress steps that communicate with NI hardware devices, or you can import LabVIEW VIs to communicate with hardware devices. You also can use a sweep operation to repeatedly measure a signal with different parameter values.

Sound and Vibration Assistant Help

Use the Sound and Vibration steps to build sound and vibration applications interactively. The Sound and Vibration Assistant is an extension of the LabVIEW SignalExpress environment and includes all LabVIEW SignalExpress functionality. The Sound and Vibration Assistant offers a customized view in the Add Step menu to help you access the most common sound and vibration measurements. Sound and Vibration Assistant Steps contains further information on using Sound and Vibration Assistant.

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:

What's New in LabVIEW SignalExpress

What's New in LabVIEW SignalExpress 3.0

LabVIEW SignalExpress 3.0 includes the following changes from LabVIEW SignalExpress 2.5.

Environment Enhancements

LabVIEW SignalExpress 3.0 includes the following environment enhancements.

General Environment Enhancements

The LabVIEW SignalExpress environment includes the following general enhancements:

- Drag-and-drop layout arrangement—You can use drag-and-drop functionality to customize the appearance of LabVIEW SignalExpress by moving views, including the Project View, Channel View, context help, and tabs.
- Auto hide functionality—You can specify to hide views when you move the cursor out of the view.

Data View Tab Enhancements

The Data View tab includes the following enhancements:

- Display properties—You can use the Display Properties dialog box to configure properties for all types of displays. This dialog box replaces the **Waveform Graph Properties** dialog box in LabVIEW SignalExpress 2.5 and earlier. The display properties you can configure depend on the format of the display and can include titles, format and precision, plots, scales, cursors, signal order, and advanced settings. Click the **Properties** button on the **Data View** tab or right-click a display and select **Properties** from the shortcut menu to display the **Display Properties** dialog box.
- Display titles—Displays appear with titles by default. You can use the Title page of the **Display Properties** dialog box to edit or hide display titles.
- Graph palette—The graph palette replaces the zoom buttons on graph displays. The graph palette also appears with graphs you add to the Project Documentation tab.
- Cursor enhancements—You can link cursors on the same display or between displays, export measurement values from the cursor legend to the Project View so you can use the value as a step input, and bind cursor measurement values to step parameters so the parameter to which you bind a cursor measurement updates to use that value. You also can define a peak threshold

and width for each cursor on a display. Use the Cursors page of the **Display Properties** dialog box to configure cursors.

Project View Enhancements

In LabVIEW SignalExpress 3.0, the Project View displays steps enclosed in execution loops. An execution loop encloses the steps in an execution group, which is a group of steps that send signals to and receive signals from each other. If you add a step that does not receive a signal from previous steps, the new step appears enclosed in a new execution loop. Steps in separate execution groups do not send signals to or receive signals from each other when you run the project. Execution loops replace the execution separators that appear in LabVIEW SignalExpress 2.5 and earlier.

Channel View Enhancements

The Channel View includes the following enhancements:

- **Shared Variables** mode—You can use the Channel View to view shared variables on the local machine or the local network. Select **Shared Variables** from the **View** pull-down menu at the top of the Channel View to display the available shared variables. When you place a checkmark in the **Acquire** checkbox for a shared variable, LabVIEW SignalExpress automatically configures a Read Shared Variables step and adds the step to the Project View.
- Import Channel View from Excel—If you export the contents of the Channel View to Microsoft Excel, you can import the contents of the Microsoft Excel spreadsheet back into the Channel View. Use this feature to share Channel View configurations across projects or to update the Channel View if you modify the information in the Microsoft Excel spreadsheet.

Project Documentation Tab Enhancements

The Project Documentation tab includes the following enhancements:

- The following toolbar buttons appear on the tab: **Print Documentation**, **Select Font**, **Left**, **Centered**, and **Right**. Use these buttons to print the contents of the tab, configure font settings, and set paragraph alignment.
- You can undo or redo operations on the tab using the Edit menu or keyboard shortcuts.

- Objects you place on the tab appear with handles you can use to resize the object.
- You can select to view the contents of the tab as they appear on a printed page or a Web page using the Documentation menu.

Run Mode Enhancements

You can configure LabVIEW SignalExpress to run a project for a number of iterations you specify or for an amount of time you specify. By default, LabVIEW SignalExpress runs projects continuously when you click the **Run** button. Click the down arrow on the **Run** button and select **Run Continuously**, **Run Once**, or **Configure Run** to specify the run mode you want to use. You also can access run mode configuration options from the Operate menu.

When you select **Configure Run**, the Configure Run dialog box appears. Use this dialog box to specify a number of iterations or a time in seconds for which you want to run the project. You also can specify whether to create a snapshot of all the signals in the project when the project finishes running. When you save a project, LabVIEW SignalExpress also saves the run mode configuration.

Note In a Playback work area, you only can run projects continuously.

Logging Enhancements

LabVIEW SignalExpress 3.0 includes the following data logging enhancements.

Event Detection

Use the Events page of the Recording Options tab to configure events that you want LabVIEW SignalExpress to acknowledge during logging. You can configure keystroke events or signal-based events. LabVIEW SignalExpress saves events in the log file, and you can specify an annotation to appear where the event occurs on the graph of the logged signal. You also can specify to prompt the user to enter an annotation when a keystroke event occurs.

Right-click a signal in the Logged Data window and select **Show Alarms and Events** from the shortcut menu to display a record of all the events and alarms that occurred during logging.

Start and Stop Condition Enhancements

Use the Start Conditions and Stop Conditions pages of the **Recording Options** tab to configure start and stop conditions. The **Start Conditions** and **Stop Conditions** pages include the following new or enhanced options:

- **Condition type**—You can specify **Software trigger** as a start or stop condition.
- **Condition logic**—If you specify multiple start or stop conditions, you also can specify the logic LabVIEW SignalExpress uses to process the conditions. You can acknowledge the start or stop conditions when all conditions occur, when one condition occurs, or when all the conditions occur in a specific order.
- **Count**—You can specify the number of times a start or stop condition must occur before LabVIEW SignalExpress acknowledges the start or stop condition.
- **Schedule start time**—You can configure repeating **Date/Time** start conditions. You can schedule LabVIEW SignalExpress to start logging weekly, daily, hourly, in smaller time increments, or on a custom schedule you specify.
- **Restart behavior**—You can configure how many times LabVIEW SignalExpress restarts logging after start and stop conditions

occur. You can configure logging to restart a number of times you specify or until a time and date you specify. You also can specify whether to save logged data in the same log file or in a new file after a restart occurs.

- **Holdoff**—You can specify an amount of time that LabVIEW SignalExpress waits after a start or stop condition occurs before acknowledging a new start or stop condition.
- **Post-stop condition duration**—You can specify a number of seconds of data to include in a log file after a stop condition occurs.

Alarm Enhancements

Use the Alarms page of the **Recording Options** tab to configure alarms. In addition to actions such as displaying a message or producing a sound when an alarm occurs, you now can generate a software trigger, create a snapshot, or run a program.

Logging Frequency-Domain Signals

You can log the last known value of a frequency-domain signal and view the log on the **Data View** tab. The last known value of the signal is the value of the signal when the project stops running. In the Project View, right-click a frequency-domain output signal and select **Record last value** from the shortcut menu to log the last known value of the signal. LabVIEW SignalExpress saves logs of frequency-domain signals in the .tdms file format to the default location you specify on the Logging page of the Options dialog box.

Note Because LabVIEW SignalExpress records a frequency-domain signal as a single value, you cannot play back a log of a frequency-domain signal.

Operator Mode

Operator mode provides a way to disable editing capabilities for LabVIEW SignalExpress projects. Disabling editing capabilities can be useful if you want to distribute a project to other users. For example, if you create a project that generates a signal, you can configure the project so that in operator mode a user only can adjust the frequency of the signal. You also can set a password on a project so that a user must enter the password to disable operator mode. If you save a project in operator mode, the project always opens in operator mode unless you save the project again with operator mode disabled.

Use the Operator Interface view and the Toolbox and Properties windows to configure the controls a user can access when a project is in operator mode. Use the Operate menu or the toolbar buttons on the Operator Interface view to enable or disable operator mode and to set a password on a project.

Shared Variables

You can write signals to shared variables, which are application-independent software items that enable you to send data across projects or across a network. Right-click a step input or output signal and select **Write to Shared Variable** from the shortcut menu to write the signal to a shared variable. You also can use the Channel View to see the shared variables that exist on the local machine or the local network.

Exporting Project Settings

You can export the configuration settings of a LabVIEW SignalExpress project to an XML file. You can use the XML file as a record of the project configuration at the time you export the settings. The XML file displays all the steps in the project, the values of every parameter in those steps, and lists any environment elements, such as tabs, work areas, and active logs, that LabVIEW SignalExpress has loaded at the time you export the project settings. Use the File menu to export project settings to XML. You can view the file in any text or XML editor.

Note The XML file is for record-keeping purposes only. You cannot use the XML file to import project settings to LabVIEW SignalExpress.

New Step

The following new step appears in LabVIEW SignalExpress 3.0:

- Sequence step—Pauses and resumes execution of steps in a project based on the configuration you specify. The Sequence step can pause the execution of a step without stopping the execution of the entire project. Because the Sequence step can pause the execution of other steps, you can use the Sequence step to allow multiple steps in the same project to use the same hardware.

What's New in LabVIEW SignalExpress 2.5

LabVIEW SignalExpress 2.5 includes the following changes from LabVIEW SignalExpress 2.0.

Project Analyzer

The Project Analyzer is a tool that analyzes your LabVIEW SignalExpress project and returns any errors, warnings, incompatibilities, or other issues in the Error List window. The Project Analyzer determines the task you want to complete and returns potential issues with the current LabVIEW SignalExpress configuration that can prevent the project from executing properly.

New Step

The following step has been added:

- Create Digital Signal step

Step Changes

LabVIEW SignalExpress includes the following changes to existing steps:

- Create Signal—The Create Signal step is now called the Create Analog Signal step.
- Software Trigger—The Software Trigger step is now called the Trigger step.

What's New in LabVIEW SignalExpress 2.0

LabVIEW SignalExpress 2.0 included the following changes from LabVIEW SignalExpress 1.1.

Data Logging

Use the integrated data logging tool to record, save, and analyze your measurements. You can record any step output. You also can analyze and process logged data by playing it through analysis steps.

Work Areas

Use work areas to perform multiple LabVIEW SignalExpress operations from within the same project.

New Steps

The following steps have been added:

- Read Shared Variables step
- Statistics step
- Trigger step

Step Improvements

LabVIEW SignalExpress includes the following changes to existing steps:

- Time Averaging—The Time Averaging step now accepts scalar data as an input.
- Histogram—The Histogram step now accepts scalar data as an input.
- User step—The User step is now called the Run LabVIEW VI step. The Run LabVIEW VI step now supports LabVIEW 7.1 and later.
- Formula (Scalar)—The Formula (Scalar) step is now called the Formula step. The Formula step now accepts waveform data as an input.
- Limit Test—You now can perform actions on LabVIEW SignalExpress data depending on the result of the Limit Test step. Navigate to the **Actions** tab of the Limit Test configuration view to configure which actions LabVIEW SignalExpress takes when a signal fails or passes a limit test.

Step Removals

LabVIEW SignalExpress no longer contains the following steps:

- NI-DAQmx Acquire
- NI-DAQmx Generate

Note LabVIEW SignalExpress still supports these steps for project files saved in a previous version of LabVIEW SignalExpress. Install the DAQ Assistant and use the DAQmx Acquire and DAQmx Generate steps to acquire and generate signals from DAQ devices.

Channel View

The Channel View provides an integrated environment for viewing and configuring hardware channel information.

DAQ Assistant Integration

LabVIEW SignalExpress now supports full DAQ Assistant integration. The DAQ Assistant dialog boxes are fully integrated in the LabVIEW SignalExpress environment. When you install the DAQ Assistant, use the DAQmx Acquire and DAQmx Generate steps to acquire and generate signals from DAQ devices.

Grouping

You can use data grouping to analyze multiple data channels at once.

New Data Viewers

Several new data viewers are available for viewing scalar and time-domain data. Right-click the Data View and select **View As** to choose the display to use for your data. LabVIEW SignalExpress only displays the data viewers available for the specified data type.

Project Documentation Tab

You can create project descriptions, display acquired data, or document your measurement results in the **Project Documentation** tab.

Options Dialog Box

Select **Tools»Options** to display the **Options** dialog box. Use the **Options** dialog box to configure various LabVIEW SignalExpress options.

Snapshots

Snapshots allow you to save a record of the current values of all step outputs in your project.

Activating Your Software

How do I activate my software?

Use the NI Activation Wizard to obtain an activation code for your software. You can launch the NI Activation Wizard two ways:

- Launch the product and choose to activate your software from the list of options presented.
- Launch NI License Manager by selecting **Start»All Programs»National Instruments»NI License Manager**. Click the **Activate** button in the toolbar.

Note If your software is a part of a Volume License Agreement (VLA), contact your VLA administrator for installation and activation instructions.

What is activation?

Activation is the process of obtaining an activation code to enable your software to run on your computer. An *activation code* is an alphanumeric string that verifies the software, version, and computer ID to enable features on your computer. Activation codes are unique and are valid on only one computer.

What is the NI Activation Wizard?

The NI Activation Wizard is a part of NI License Manager that steps you through the process of enabling software to run on your machine.

What information do I need to activate?

You need your product serial number, user name, and organization. The NI Activation Wizard determines the rest of the information. Certain activation methods may require additional information for delivery. This information is used only to activate your product. Complete disclosure of National Instruments licensing privacy policy is available at ni.com/activate/privacy. If you optionally choose to register your software, your information is protected under the National Instruments privacy policy, available at ni.com/privacy.

How do I find my product serial number?

Your serial number uniquely identifies your purchase of NI software. You can find your serial number on the Certificate of Ownership included in your software kit. If your software kit does not include a Certificate of Ownership, you can find your serial number on the product packing slip or on the shipping label.

If you have installed a previous version using your serial number, you can find the serial number by selecting the **Help»About** menu item within the application or by selecting your product within NI License Manager (**Start»All Programs»National Instruments»NI License Manager**). You can also contact your local National Instruments branch.

What is a Computer ID?

The computer ID contains unique information about your computer. National Instruments requires this information to enable your software. You can find your computer ID through the NI Activation Wizard or by using NI License Manager, as follows:

1. Launch NI License Manager by selecting **Start»All Programs»National Instruments»NI License Manager**.
2. Click the **Display Computer Information** button in the toolbar.

For more information about product activation and licensing refer to ni.com/activate.

Related Documentation

The Getting Started with LabVIEW SignalExpress manual is a Portable Document Format (PDF) file. You must have Adobe Reader 6.0.1 or later installed to view the PDF. Refer to the Adobe Systems Incorporated Web site to download Adobe Reader. Refer to the National Instruments Product Manuals Library for updated documentation resources. Depending on the LabVIEW SignalExpress supported applications you have installed, the **Help** menu contains links to the user manual for each application.

The following documents contain information that you might find helpful as you use LabVIEW SignalExpress:

- *NI-DAQmx Help*—This help file contains information about tasks, channels, and other NI-DAQmx concepts.
- *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.

LabVIEW SignalExpress LE

LabVIEW SignalExpress LE is a free, limited version of LabVIEW SignalExpress that does not include all of the steps or features available in the Full Edition. When you install LabVIEW SignalExpress, you have a 30-day trial of the Full Edition of LabVIEW SignalExpress. After that period, you must activate LabVIEW SignalExpress LE or purchase the Full Edition. If you activate LabVIEW SignalExpress LE before the 30-day trial of the Full Edition is complete, you still can use the full features for the remainder of the 30-day trial.

LabVIEW SignalExpress LE Features

By default, LabVIEW SignalExpress LE displays all steps, including steps that are only available in the Full Edition. You can configure LabVIEW SignalExpress not to display steps available in the Full Edition. Select **Tools»Options** to display the **Options** dialog box. On the **General** page, select **No** from the **Show unlicensed steps** option.

The following table displays the features available in LabVIEW SignalExpress LE and LabVIEW SignalExpress Full Edition:

	LabVIEW Signal Express LE	**LabVIEW SignalExpress Full Edition**
Instrument Support		
Over 300 common standalone instruments	✓	✓
Visualization and Documentation		
Customizable graphing	✓	✓
Interactive cursors	✓	✓
Save signals to file	✓	✓
Print and export graphs	✓	✓
Drag and drop data into Microsoft Excel, Word, and WordPad	✓	✓
Operator mode with limited user-editing	⊘	✓
Signal Processing		
Software filters	⊘	✓
Scalar and waveform math	⊘	✓
Analog and digital conversion	⊘	✓
Interactive signal comparisons	⊘	✓
Load simulation data from PSPICE, Multisim, and other SPICE packages	⊘	✓
Time and Frequency Measurements		

Amplitude and level	⊘	✓
Timing and transition	⊘	✓
Power spectrum	⊘	✓
Frequency response	⊘	✓
Distortion measurements	⊘	✓
Tone extraction	⊘	✓
Data Logging		
Limited data logging (one log per project)	✓	⊘
Unlimited data logging	⊘	✓
Logging alarms and events	⊘	✓
Logging with start and stop conditions	⊘	✓
Measurement Automation		
Parameter sweeping	⊘	✓
Limit testing	⊘	✓
Software triggering	⊘	✓
Sequencing	⊘	✓
Remote Data Access		
Read/Write shared variables	⊘	✓

How do I get help?

LabVIEW SignalExpress includes a dynamic context help window that displays step, tab, and parameter-level help. Move the cursor over an object to display context help for that object.

You also can select **Help»LabVIEW SignalExpress Help** to access the *LabVIEW SignalExpress Help*.

Using Help

Conventions

This help file uses the following formatting and typographical conventions:

< >	Angle brackets that contain numbers separated by an ellipsis represent a range of values associated with a bit or signal name—for example, AO <0..3>.
[ ]	Square brackets enclose optional items—for example, [response].
»	The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
	This icon denotes a tip, which alerts you to advisory information.
	This icon denotes a note, which alerts you to important information.
blue	Text in this color denotes a specific platform and indicates that the text following it applies only to that platform.
bold	Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.
green	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.
`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.

monospace bold	Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.
monospace italic	Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Navigating Help

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

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

Wildcards

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

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

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

Nested Expressions

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

Boolean Expressions

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

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

Search Options

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

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

Printing Help File Topics

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.

Navigating the LabVIEW SignalExpress Environment

The LabVIEW SignalExpress environment consists of views that display various types of information. The primary view appears in the middle of the application window and contains tabs in the default layout. Each tab has a unique functionality, such as displaying data, configuring steps, or documenting projects. The primary view is surrounded by supplementary views that, unless you close the views, are always visible. Supplementary views include the Project View, which you use to build projects, the Context Help window, which displays information about tabs and steps, and the Channel View, which displays configurable items such as hardware devices and shared variables. The Toolbox and Properties windows also are supplementary views.

Note When you first launch LabVIEW SignalExpress, the Data View tab appears in the primary view, and the Project View and the **Context Help** window appear as supplementary views. If LabVIEW SignalExpress detects hardware devices or shared variables, the Channel View also appears. However, you can customize the LabVIEW SignalExpress environment to display any view in any location.

Menus and toolbar buttons appear across the top of the LabVIEW SignalExpress environment. Use the menus and toolbar buttons to manage files, access steps, run projects, record logs, modify the environment, and access help. For example, you can use the View menu to display views in LabVIEW SignalExpress.

Views

The LabVIEW SignalExpress environment consists of views that display various types of information. The primary view appears in the middle of the application window and is usually the largest view. In the default layout, the primary view contains tabs, such as the Data View tab. Supplementary views surround the primary view and appear with individual title bars. For example, the Project View is a supplementary view.

You can customize the appearance of the LabVIEW SignalExpress environment by selecting which views to display, specifying how LabVIEW SignalExpress displays the views, and arranging the views in the application window.

View Display Options

To display a view, select the view from the View menu. After you display a view, you can choose one of the following methods for displaying the view:

- Floating—Displays the view in a separate, floating window.
- Hiding—Hides the view when you move the cursor out of the view. LabVIEW SignalExpress places buttons on the edge of the application window to indicate a hidden view. Move the cursor over the button to display the hidden view.
- Docking—Docks the view in a fixed location.

Displaying Views

You can customize the appearance of the LabVIEW SignalExpress environment by selecting views to display, specifying how LabVIEW SignalExpress displays the views, and arranging the views in the application window. Select a view from the View menu to display the view.

Floating, Hiding, and Closing Views

The method you use to float, hide, dock, or close a view depends on whether the view appears as a primary view or a supplementary view.

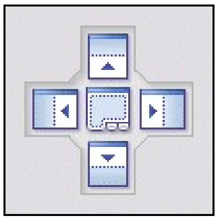
Note With the exception of the Data View tab, closed views retain any information you enter prior to closing the view. For example, closing the Project View does not remove the steps in the project.

If a view appears as a tab in the primary view area, you can right-click the name of the tab to display a shortcut menu of display options. Select **Float** to make the tab a floating view or **Close** to close the tab. You also can click the **X**, or **Close** button, in the primary view to close a tab. After you make a tab a floating view, you can drag the view over the docking icons to dock the view.

If a view appears as a supplementary view with a title bar, you can click the down arrow on the title bar to display a drop-down menu of display options. The **Float** and **Close** options work in the same way for primary and supplementary views. Supplementary views also include an **Auto Hide** option, which you can use to hide the view when you move the cursor out of the view. You also can use the **Auto Hide** and **X**, or **Close** buttons on the title bar of a supplementary view to hide or close the view.

Dragging and Docking Views

You can drag a floating view or a view with a title bar over a docking icon to dock the view in a new location. The following illustration shows the docking icons that appear near the middle of the application window when you begin dragging a view.



Drag a view over a docking icon to see where the view appears if you drop the view. If you drag a view over the middle, tabbed icon, the view appears tabbed when you drop the view.

Resetting the Environment Layout

Select **View»Reset Layout** to reset the LabVIEW SignalExpress environment to the default layout.

Note Resetting the layout closes any views that do not appear in the default layout. With the exception of the Data View tab, closed views retain any information you enter prior to resetting the layout.

Project View

The Project View displays the functional steps of a LabVIEW SignalExpress project. You create projects by adding steps to the Project View.

Steps

Steps are functions you can configure to acquire, analyze, save, or load signals in the project. Each step consists of input signals and/or output signals.

- Step input signals are signals you pass into a step that you want to analyze, process, generate, or save. Step input signals appear as red arrows on a step.
- Step output signals are signals a step creates, acquires, analyzes, processes, or imports and returns. You can use step output signals as inputs to subsequent steps. Step output signals appear as blue arrows on a step. You can drag an output signal to the Data View or use a probe to view the data the output signal holds.

You must add steps to the Project View in a logical execution sequence. If a step requires an input signal from another step, the step providing the input must appear before the step that requires the input. If you add a step that cannot find an input it needs or that cannot process the data a preceding step or steps return, an error message appears that explains where the error occurred and suggests ways to fix the error.

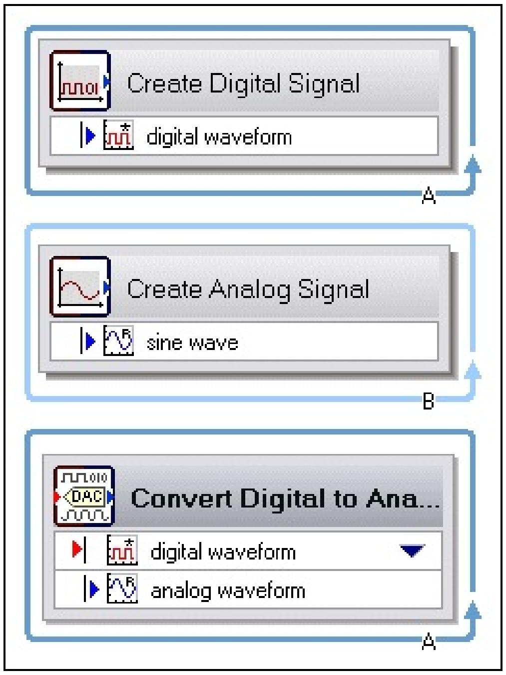
Execution Loops

Execution loops are blue loops that show the execution of steps in the Project View. An execution loop encloses the steps in an execution group, which is a group of steps that send signals to and receive signals from each other. If you add a step that does not receive a signal from previous steps, the new step is part of a new execution group and appears enclosed in a new execution loop. Steps in separate execution groups do not send signals to or receive signals from each other when you run the project.

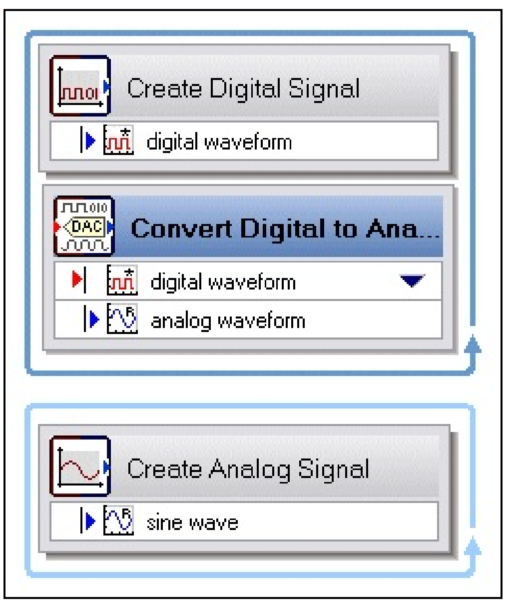
LabVIEW SignalExpress determines execution groups automatically based on the order in which steps appear in the Project View. Steps that provide output signals must appear above steps that require those signals as inputs. Because steps in one execution group do not have dependencies on steps in other execution groups, separate execution groups execute independently of each other.

If you add a step to the Project View or drag a step to a position where the step can receive signals from some preceding steps but not others, the step appears enclosed in a new execution loop, and letters appear in the bottom right corner of each execution loop in the project. The letters indicate which steps are in the same execution group.

The following illustration shows execution loops with letters. The execution loops that show the letter **A** contain steps that execute as part of the same group when you run the project. The execution loop that shows the letter **B** executes independently.



If an execution loop becomes divided into two loops, you can combine the loops by dragging steps between the sections of the execution loop. The following illustration shows the same project after you drag the step from the second **A** loop to the first **A** loop. Letters appear on execution loops only when the number of execution loops in the Project View exceeds the actual number of execution groups.



LabVIEW SignalExpress always draws a new execution loop if you drag a step that requires an input signal to a location where it no longer can determine which input signal to use. When you specify the input signal you want to use, LabVIEW SignalExpress automatically moves the step into the appropriate execution loop.

Configuring the Project View

You can use the **Options** dialog box to configure the Project View to show or hide the status bar, show or hide inputs and outputs of steps, and display large or small icons in steps.

Tabs

Tabs appear in the primary view in the default LabVIEW SignalExpress environment layout. Use the tabs to display data, configure steps and operations, document projects, and view errors, warnings and notifications.

Each tab has a specific functionality, and you do not need to use every tab for every project. Select a tab from the View menu to display the tab. Click the **X**, or **Close** button, or right-click a tab and select **Close** from the shortcut menu to close a tab.

Note With the exception of the Data View tab, closed tabs retain any information you enter prior to closing the tab. For example, closing the Project Documentation tab does not erase the documentation you enter on the tab.

LabVIEW SignalExpress includes the following tabs:

- Event Log—Displays events such as errors, warnings, and alarms that occur in LabVIEW SignalExpress. For each event, the **Event Log** tab displays the severity of the event and the time the event occurred, as well as the source and title of the event, if known.
- **Data View**—Displays signals and data from steps in graphs, charts, and other display formats.
- Recording Options—Configures logging operations.
- **Project Documentation**—Displays documentation that you create for the current project. You can enter text, import images, and drag signals from the Project View to the **Project Documentation** tab.
- Step Setup—Configures steps in the project.

Note Some programs, such as NI-DAQmx, install additional tabs.

Context Help

The **Context Help** window displays information about views and steps and appears on the right side of the LabVIEW SignalExpress application window in the default layout.

The **Context Help** window is split into two sections. The top section displays basic information about a view or step when you move the cursor into the view or the Step Setup tab. The bottom section displays parameter-specific information when you move the cursor over a parameter on the **Step Setup** tab for a step.

If the **Context Help** window is not visible, select **Help»Context Help** to display the **Context Help** window.

Add Step Palette

The **Add Step** palette contains the steps you use to build LabVIEW SignalExpress projects. The steps appear in subpalettes based on the functionality of the step.

Click the **Add Step** button to display the **Add Step** palette. A temporary version of the palette appears. If you click outside of the **Add Step** palette, the palette disappears. Click the thumbtack in the upper left corner of the palette to pin the palette so it is no longer temporary.

The order of steps in the **Add Step** palette is identical to the order of steps in the Add Step menu. You can use either the palette or the menu to add steps to a project.

Using Projects

LabVIEW SignalExpress projects are collections of steps that can acquire, analyze, create, generate, display, and log signals. The functionality of a project depends on the steps you use to build the project. For example, you can use the IVI Scope Acquire step and the Filter step to create a project that acquires a signal from an oscilloscope and filters the signal.

Select **File»New Project** to create a new, empty project. You then can build the project by adding steps to the Project View. You also can create multiple work areas to perform multiple LabVIEW SignalExpress operations from within the same project. When you save the project, LabVIEW SignalExpress saves all the work areas in the project. LabVIEW SignalExpress saves projects with a .seproj file extension.

LabVIEW SignalExpress also saves the environment layout for a project. For example, if you save a project with only the Data View and Event Log tabs visible, the Data View and Event Log appear by default the next time you open the project. However, you can modify the LabVIEW SignalExpress environment at any time.

Steps

Steps are functions you can configure to acquire, analyze, save, or load signals in a project. Each step consists of input signals and/or output signals, and has a specific functionality. For example, the Filter step filters a signal using an infinite impulse response (IIR) or finite impulse response (FIR) filter, and the Time Averaging step averages a time signal or scalar input.

The following image shows the Create Analog Signal step as it appears in the Project View.

The arrows that appear on a step indicate whether the step processes input signals, generates output signals, or does both. The blue arrow in the previous image indicates that the Create Analog Signal step generates an output signal.

- Step input signals are signals you pass into a step that you want to analyze, process, generate, or save. Input signals appear as red arrows on a step in the Project View.
- Step output signals are signals a step creates, acquires, analyzes, processes, or imports and returns. You can use output signals as inputs to subsequent steps. Output signals appear as blue arrows on a step in the Project View. You can drag an output signal to the Data View or use a probe to view the data the output signal holds.

Data Type icons denote the signal type of step input signals and step output signals.

Use the Add Step palette or the Add Step menu to add steps to a project.

Configuring Steps

Every LabVIEW SignalExpress step has a basic functionality, but you can configure the step to specify the input signals it processes, the output signals it creates, or the way in which it performs a specific action. For example, you must specify the type of filter you want the Filter step to use.

Use the Step Setup tab to configure a step. The **Step Setup** tab appears automatically when you add a step to the Project View.

Resetting or Restarting Steps

Some steps perform processes that depend on multiple iterations. These steps must run in Run Continuous mode to provide accurate results. For example, the Filter step requires some settling time to acquire enough data to filter a signal correctly. The amount of time these steps must run to provide accurate results depends on the input signal of the step.

Steps that require multiple iterations to produce accurate results appear with a **Reset** or **Restart** button in the toolbar of the Step Setup tab. Clicking the **Reset** or **Restart** button resets the step to its initial state and restarts the processes of the step.

Adding and Deleting Steps

Click the **Add Step** button on the toolbar and select a step from the **Add Step** palette to add a step to the Project View. You also can right-click the Project View and select a step from the shortcut menu or select a step from the **Add Step** menu.

After you add a step to the Project View, you can right-click the step and select **Insert Before** or **Insert After** from the shortcut menu to add another step to the Project View.

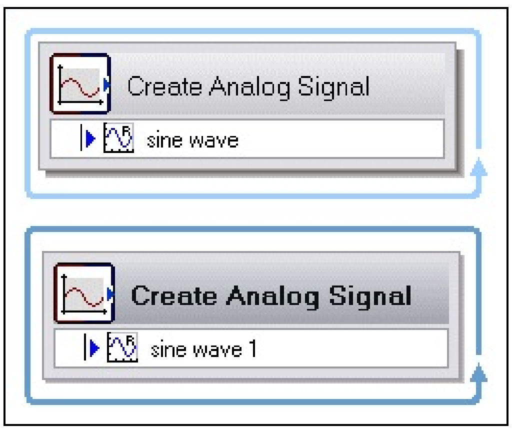
You also can add a step to the Project View by right-clicking an output signal on a step and selecting **Send To** from the shortcut menu. The shortcut menu displays a list of steps that can handle the output signal. When you select a step for the output signal, LabVIEW SignalExpress places the step in the Project View below the step you originally right-clicked.

To remove a step from the Project View, select the step and press the <Delete> key or right-click the step and select **Delete** from the shortcut menu.

Step Execution Groups

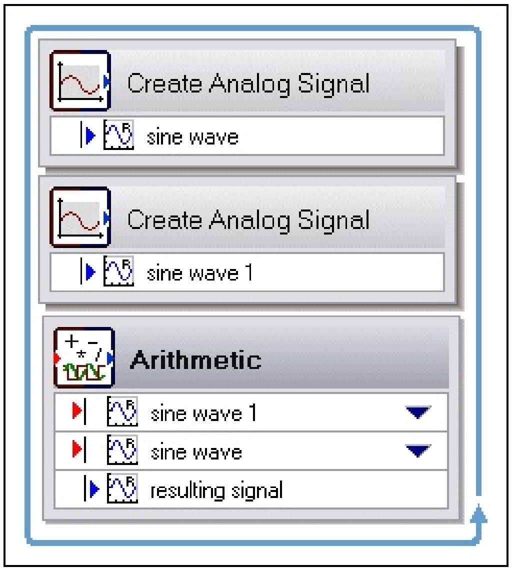
In a project, steps that send signals to and receive signals from each other execute together as an execution group. The timing of the execution of individual steps in an execution group depends on factors such as signal inheritance and hardware configuration. For example, a step that analyzes a signal cannot execute until it receives the signal from a previous step or an instrument. Steps in an execution group appear enclosed in a blue execution loop in the Project View.

You can create projects where two or more groups of steps execute independently of each other. Steps in one execution group do not receive signals from steps in other execution groups. For example, if you add two Create Analog Signal steps to the Project View, the steps appear in separate execution loops, as shown in the following image, because the second Create Analog Signal step does not receive a signal from the first Create Analog Signal step.



When you run the project, both Create Analog Signal steps execute concurrently because one step does not need to wait on the other step.

LabVIEW SignalExpress determines whether steps can execute in the same group automatically. For example, if you modify the previous example to add an Arithmetic step after the second Create Analog Signal step, all three steps appear in one loop, as shown in the following illustration, because the Arithmetic step receives data from both Create Analog Signal steps.



When you run the project, both Create Analog Signal steps execute, then the Arithmetic step executes.

If you add two more Create Analog Signal steps and another Arithmetic step after the first Arithmetic step, the new steps appear in a separate execution loop because the second group does not receive data from the first group. The two groups execute independently when you run the project. Common uses for multiple execution groups in the same project include creating multi-rate applications and simultaneously sending two different signals to two different hardware devices running at two different speeds.

You also can create projects where some steps run independently of others in the same execution group. For example, if you add a Create Analog Signal step, a Filter step, and a Scaling and Conversion step to the Project View and configure both the Filter step and the Scaling and Conversion step to use the output signal of the Create Analog Signal step as an input, the Filter step and the Scaling and Conversion step execute concurrently. The Scaling and Conversion step does not have to wait for the Filter step to finish executing. Both steps execute immediately after the Create Analog Signal step executes.

If you configure an independently executing step incorrectly, an error occurs at that step, and the step stops executing and does not return a signal. However, other steps that are not dependent on that step for a signal continue to execute.

Acquiring Signals

Most projects you create using LabVIEW SignalExpress analyze or log signals you acquire from a data acquisition device or modular instrument. LabVIEW SignalExpress provides the following methods for acquiring signals:

- Acquiring Signals Using USB Plug-and-Play—If you use LabVIEW SignalExpress with an NI USB device, such as NI CompactDAQ, the NI Device Monitor that installs with NI-DAQmx software detects when you connect the device to the PC. Use USB plug-and-play functionality if you want LabVIEW SignalExpress to configure data acquisition and begin acquiring signals automatically.
- Acquiring Signals Using the Channel View—If you use LabVIEW SignalExpress with NI-DAQmx or NI-DMM devices or NI Switch modules, the Channel View displays all analog input (AI) channels available for the device or module. You can use the Channel View to acquire signals from the supported devices.

Note You can install NI-DAQmx, NI-DMM, or NI Switch software from the National Instruments Device Driver DVD, or you can download the latest version of NI-DAQmx, NI-DMM, or NI Switch software from ni.com/downloads.

- Acquiring Signals Using Steps—You can use the Acquire Signals steps to acquire signals from a hardware device. You can use the IVI steps to acquire signals from hardware of the specified IVI class. Some NI modular instruments, such as NI-DMM, NI-SCOPE, and NI-HSDIO also install instrument-specific Acquire Signals steps.

Acquiring Signals Using USB Plug-and-Play

If you use LabVIEW SignalExpress with an NI USB device, such as NI CompactDAQ, the NI Device Monitor detects when you connect the device to the PC. LabVIEW SignalExpress then can set up a project automatically and begin acquiring signals from the device with minimal user input.

Note The NI Device Monitor installs with NI-DAQmx. You can install NI-DAQmx software from the National Instruments Device Driver DVD, or you can download the latest version of NI-DAQmx software from ni.com/downloads. After you install NI-DAQmx, refer to the *NI-DAQmx Help* for more information about using NI-DAQmx.

Complete the following steps to acquire signals from an NI USB device using LabVIEW SignalExpress.

1. Use a USB cable to connect the device to the PC.
2. Turn on the device.
3. (Optional) If you have not already done so, install any necessary software for the device. Windows Vista installs device-specific software automatically when it detects a new device. On Windows XP and earlier, the Found New Hardware wizard prompts you to install the software.
4. In the **New Data Acquisition Device** dialog box that appears, select **Begin a Measurement with This Device Using NI LabVIEW SignalExpress** and click the **OK** button. A **LabVIEW SignalExpress** window with an image of the device appears.

   You can add modules to the device at this time and the image automatically updates to display the new modules. Refer to the device documentation for more information about the modules you can use with a specific device.
5. Select the modules from which to acquire signals and click the **OK** button. LabVIEW SignalExpress launches and runs a project that acquires signals from the modules you selected.

Note If you select modules that generate signals, such as analog output (AO) modules, LabVIEW SignalExpress prompts you for input signals to use for the generated signals.

Acquiring Signals Using the Channel View

If you use LabVIEW SignalExpress with installed or simulated NI-DAQmx devices or with an NI-DMM device or NI Switch module, you can use the Channel View to acquire signals from all analog input (AI) channels available for the devices or modules. LabVIEW SignalExpress automatically detects all the devices the Channel View supports and displays them in the Channel View. You can install NI-DAQmx, NI-DMM, or NI Switch software from the National Instruments Device Driver DVD, or you can download the latest version of NI-DAQmx, NI-DMM, or NI Switch software from ni.com/downloads.

Note The Channel View does not display analog output (AO) channels.

Complete the following steps to acquire signals from a supported device or module using the Channel View.

1. Install the device and connect it to the PC, or use Measurement and Automation Explorer (MAX) to create a simulated device. Refer to the device documentation for more information about installing a device.

Note MAX installs with most National Instruments software applications. Refer to the *Measurement & Automation Explorer Help* for more information about creating NI-DAQmx simulated devices.

2. In LabVIEW SignalExpress, select **View»Channel View** to display the Channel View. LabVIEW SignalExpress displays the devices it detects in the **Physical Channel** column. If you do not see the device you want to use, click the **Refresh** button to update the Channel View.
3. Click the expand symbol for the device from which you want to acquire signals to display the device channels.
4. Place a checkmark in the **Acquire** checkbox for each channel from which you want to acquire a signal. If you want to acquire signals from all channels, place a checkmark in the **Acquire** checkbox for the device itself.

   When you place a checkmark in an **Acquire** checkbox, additional measurement configuration options appear in the Channel View table, and LabVIEW SignalExpress automatically places a data acquisition step in the Project View. The measurement configuration options that appear depend on the device you use.

5. (Optional) Use the measurement configuration options to configure settings for the acquisition. For example, for most devices you can specify the **Measurement Type** and **Scaled Units** to use for the measurement. You also can use the **Properties** window to configure multiple channels simultaneously.
6. Drag the outputs of the data acquisition step to the Data View and click the **Run** button to run the project. LabVIEW SignalExpress acquires signals from the device and displays the signals in the Data View.

Acquiring and Generating Signals Using Steps

You can use the Acquire Signals steps and the Generate Signals steps to acquire or generate signals from a hardware device. You can use the IVI steps to acquire or generate signals from hardware of the specified IVI class. Some NI modular instruments, such as NI-DMM, NI-SCOPE, and NI-HSDIO also install instrument-specific Acquire Signals and Generate Signals steps.

Use steps to acquire signals when you are using a device that does not support USB plug-and-play signal acquisition or that LabVIEW SignalExpress cannot detect automatically using the Channel View. You also can use steps when you want to use a device to generate a signal. For example, if you want to generate a signal from an analog output (AO) device that does not support USB plug-and-play functionality, you can use the Generate Signals steps to configure the device. Complete the following steps to acquire or generate a signal using steps.

1. Install the device and connect it to the PC, or use Measurement and Automation Explorer (MAX) to create a simulated device. Refer to the device documentation for more information about installing a device.

Note MAX installs with most National Instruments software applications. Refer to the *Measurement & Automation Explorer Help* for more information about creating NI-DAQmx simulated devices.

2. Select the appropriate step from the Add Step palette or the Add Step menu to add the step to the Project View.
3. Use the Step Setup tab to configure the Acquire Signals or Generate Signals step.
4. (Optional) If you select a Generate Signals step, add a step to the Project View before the Generate Signals step to provide the form of the signal you want to generate. For example, you can use the Create Analog Signal step to generate a simple sine wave or other signal.
5. Click the **Run** button to run the project and begin acquiring or generating the signal.

Running and Stopping Projects

When you run a project, LabVIEW SignalExpress executes all the steps that the project contains until you stop or abort the project.

Running Projects

Click the **Run** button to run a LabVIEW SignalExpress project.

LabVIEW SignalExpress prompts you to configure the run mode for a project the first time you click the **Run** button after launching LabVIEW SignalExpress. By default, LabVIEW SignalExpress runs projects continuously when you click the **Run** button. However, you can configure LabVIEW SignalExpress to run a project once, for a number of iterations you specify, or for an amount of time you specify. After you configure the run mode, the icon on the **Run** button updates to reflect the configuration.

Stopping Projects

When a project is running, the **Stop** button replaces the **Run** button. Click the **Stop** button to stop the project. When you click the **Stop** button, the project stops when the current iteration of the project completes. Projects you configure to run once, run for a specific number of iterations, or run for a specific amount of time stop automatically when the stop condition occurs.

Aborting Projects

In some cases you cannot stop a project, such as when a hardware step waits on a trigger that does not occur. In this case you must abort the project to stop execution. You can click the down arrow on the **Stop** button and select **Abort** from the shortcut menu to halt the execution of a project immediately. Because aborting a project does not wait for the final iteration of the project to complete, selecting the **Abort** option might cause LabVIEW SignalExpress to display incomplete data.

Note You also can use the options in the Operate menu to run, stop, or abort a project.

Documenting Projects

Use the Project Documentation tab to create documentation for a LabVIEW SignalExpress project. You can enter text on the tab, import images (GIF, JPG, BMP, PNG, or EMF) from other locations or drag signals from the Project View to the **Project Documentation** tab.

Displaying Signal Graphs

When you drag a signal from the Project View to the **Project Documentation** tab, the signal appears in a graph that is similar to a graph on the Data View tab. Right-click a graph on the **Project Documentation** tab to see the same shortcut menu options that appear for graphs on the **Data View** tab. For example, you can use the shortcut menu to add displays, set visible items, select signals, and change graph properties. Use the resizing handles to resize the graph on the **Project Documentation** tab.

You also can use the zoom buttons to change the portion of the signal the graph displays. Like the **Data View** tab, the **Project Documentation** tab displays the Preview Graph when you zoom in on a signal. Use the Preview Graph to select the portion of the signal to display. You can right-click the signal graph and select **Visible Items»Preview Graph** from the shortcut menu to show or hide the Preview Graph, or you can right-click the Preview Graph and select **Hide Preview** from the shortcut menu to hide the Preview Graph.

When you run the project, the signals on the **Project Documentation** tab update to display the current values of the signals while the project runs. When you print project documentation, LabVIEW SignalExpress prints the most recent value of the signal.

Use the options in the Documentation menu or the toolbar buttons on the **Project Documentation** tab to configure display options and font settings, align paragraphs, insert images, or set the size of objects on the **Project Documentation** tab.

Printing and Exporting Documentation

You can print LabVIEW SignalExpress documentation that you create on the Project Documentation tab or export the documentation to HTML. To print documentation, click the **Print Documentation** toolbar button on the **Project Documentation** tab, or select **File»Print»Print Documentation**. If the **Project Documentation** tab includes graphs of signals, LabVIEW SignalExpress prints the most recent value of the signal.

To export documentation to HTML, select **File»Export»Export Documentation to HTML**, specify the filename for the HTML file you want to create, and click the **Save** button.

Exporting Project Settings to XML

You can export the configuration settings of a LabVIEW SignalExpress project to an XML file. You can use the XML file as a record of the project configuration at the time you export the settings. Select **File»Export»Export Project Settings To XML** to export the current configuration settings of a project to an XML file. You can view the XML file in any text or XML editor.

The XML file displays all the steps in the project, the values of every parameter in those steps, and lists any environment elements, such as tabs, work areas, and active logs, that LabVIEW SignalExpress has loaded at the time you export the project settings. LabVIEW SignalExpress formats the data in the XML file so you easily can identify the value(s), name(s), and type(s) of the data from the tags that describe the data. The XML file is for record-keeping purposes only. You cannot use the XML file to import project settings to LabVIEW SignalExpress.

> **Note** The XML file displays values for every parameter of every step in the project, including parameters that you are not using or that are not currently visible. For example, some parameters appear on the Step Setup tab only when you configure a step in a particular way. However, the XML file still records values for these parameters. If you have not specified a value for a parameter, the XML file displays the default value.

When you export project settings to XML, the XML file does not include signal data such as step output values. You can use shortcut menu options or the Save to ASCII/LVM step to save output or logged data to a file.

Using National Instruments Hardware with LabVIEW SignalExpress

With LabVIEW SignalExpress, you can control NI data acquisition boards and modular instruments using the steps in the **Acquire Signals** step menu.

You can perform the following hardware operations using LabVIEW SignalExpress steps:

- Configure the basic acquisition or generation options, such as channel selection, acquisition rate, number of points, input range, and so on.
- Use triggering capabilities of the device.
- Synchronize multiple boards by sharing clock and/or trigger signals between boards.

Communicating with Instruments Using IVI

LabVIEW SignalExpress provides a set of IVI steps that communicate with instruments in the following IVI Classes:

- Digital Multimeter
- Oscilloscope
- DC Power Supply
- Arbitrary Waveform/Function Generator

To communicate with an instrument, you need to install the instrument-specific IVI driver and create a session name for the instrument.

Downloading and Installing the Instrument-Specific IVI Driver

Complete the following steps to download an IVI-specific driver for an instrument.

1. Launch the Instrument Driver Network.
2. Use the **Instrument Type** list on the left side of the page to find a driver for the instrument, or click the **Browse Drivers** link and use the **Search within results** field to search for the instrument you are using.
3. When you find a driver you want to download, click the **Model** name to open the description of the driver.
4. Confirm that IVI appears in the **Driver Type** description.
5. Download and install the driver.

Creating an IVI Session Name

Each LabVIEW SignalExpress IVI step contains an **IVI session name** control that allows you to select the session name that you want to associate with the step. The session name identifies the driver and the instrument that you want to use with the step. From the **IVI session name** control, you can select a previously created session name or create a new local session name from the step.

Complete the following steps to create an IVI logical name from the step.

1. Select **Create New** from the **IVI session name**.
2. Enter a name and select the Resource descriptor associated with the instrument that you want to control, or enter the descriptor if it does not appear in the **Resource descriptor** control.
3. Select the appropriate Instrument Driver and click the **OK** button.

Note You can simulate your hardware by placing a checkmark in the Enable simulation data checkbox.

Deployment

The IVI steps in LabVIEW SignalExpress communicate to an instrument through an IVI Session. The IVI Session is stored in the IVI Configuration Store, not in your LabVIEW SignalExpress project. Therefore, the IVI Session is not transferable between computers.

Deploying a LabVIEW SignalExpress Project

To bring your LabVIEW SignalExpress project to a computer that has LabVIEW SignalExpress installed, you can either reconfigure the IVI steps to use existing IVI Sessions, or if no appropriate IVI Sessions exist, create new IVI Sessions from within LabVIEW SignalExpress.

1. Select **Create New** from the **IVI session name**.
2. Enter a name and select the Resource descriptor associated with the instrument that you want to control, or enter the descriptor if it does not appear in the **Resource descriptor** control.
3. Select the appropriate Instrument Driver and click the **OK** button.

Note You can simulate your hardware by placing a checkmark in the Enable simulation data checkbox.

Deploying LabVIEW Code Converted from a LabVIEW SignalExpress Project

Converted code from a LabVIEW SignalExpress project uses the same IVI Session that was configured in LabVIEW SignalExpress. If you deploy your generated code to a new machine, you must recreate the IVI Session on that machine. If LabVIEW SignalExpress is installed on the machine, refer to Distributing LabVIEW Block Diagrams for Execution. If LabVIEW SignalExpress is not installed, you can create the IVI Session using National Instruments Measurement & Automation Explorer (MAX).

1. Launch MAX.
2. Expand the **IVI Drivers**.
3. Follow the instructions in the MAX category help.

You also can refer to the NI Developer Zone for more information about using MAX to configure an IVI system.

Resource Descriptor

A resource descriptor is a string, such as a VISA resource descriptor, that specifies the interface and the address of the hardware to associate with the step. The following lists examples of valid resource descriptors.

- GPIB::22::INSTR
- GPIB1::22::5::INSTR
- VXI::64::INSTR
- ASRL2::INSTR
- GPIB::22::INSTR
- DAQ::1::INSTR
- PXI1Slot2

Note The resource descriptor is not necessary if you place a checkmark in the **Enable simulation data** checkbox.

Instrument Driver

The driver list contains the list of all driver sessions to which the step can refer. By selecting a driver from the list, you are associating the session name with a particular set of properties the driver can use.

Enable simulation data

If you place a checkmark in the **Enable simulation data** checkbox, the specific driver functions simulate instrument I/O. If you remove the checkmark, the specific driver functions do not return random simulated values for output parameters that represent instrument data.

Hardware Timing and Software Timing

You can use hardware timing or software timing to control when certain actions, such as a device acquiring or generating a signal, occur in LabVIEW SignalExpress. With hardware timing, a digital signal, such as a clock on a device, controls the rate of timed actions. With software timing, LabVIEW SignalExpress determines the rate of timed actions using the operating system timer. Hardware timing usually is more accurate than software timing.

Note Some devices do not support hardware timing. Refer to the device documentation to determine if a device supports hardware timing.

LabVIEW SignalExpress also supports hardware timing options, such as sample clock sharing, that you can use to synchronize devices. Timing configuration options appear on the Step Setup tab of most Acquire Signals and Generate Signals steps.

Synchronizing Devices

In addition to the automatic synchronization of devices offered by NI-DAQmx for devices such as DSA or SMIO, you can synchronize devices by sharing trigger and sample clock signals and by sharing a master timebase or a reference clock.

Refer to the Synchronized Analog Input (Share Trigger and Timebase) example located in the SignalExpress\Examples\DAQmx directory for an example of synchronizing two devices.

Note LabVIEW SignalExpress does not support trigger, clock, timebase, or reference clock sharing across PXI Trigger buses on the NI PXI-1006 and NI PXI-1045 chassis.

Trigger Sharing

You can synchronize devices by sharing trigger signals.

For example, if you configure the start trigger of a DAQmx Acquire step with the **Trigger type** set to **Digital**, you can select any available trigger signal provided by the other devices currently in the project in the **Trigger source** pull-down menu. When you select one of these trigger signals, LabVIEW SignalExpress routes the trigger signal between the master and the slave and controls the master and slave execution. LabVIEW SignalExpress initializes the slave device before the master device so the slave device can receive the trigger signal.

Sample Clock Sharing

You can synchronize devices by sharing sample clock signals.

You can configure a DAQmx Acquire step by clicking the **Sample clock type** pull-down menu and selecting **External** on the **Advanced timing** tab. The **clock source** pull-down menu lets you select any sample clock signal provided by the other devices currently in the project. As for triggers, LabVIEW SignalExpress handles the sample clock routing and the execution control.

Acquiring and Generating Analog Signals Simultaneously

LabVIEW SignalExpress can execute steps in projects simultaneously. For example, if you have two DAQmx Acquire steps in a project, each step runs at its own rate and independently of the other when you run the project. You can use triggers or you can share data between steps to control the order in which steps execute in a project. For example, when you use LabVIEW SignalExpress to generate a stimulus signal and then measure the response from a unit under test, you must make sure that the acquisition does not occur until after the DAQmx Generate step generates the stimulus signal. The following section describes different options for controlling the execution relationship between hardware steps to ensure measurements operate appropriately.

Execution Dependencies in Hardware Steps

If a project contains two DAQmx Acquire steps and you configure the steps to share the same start trigger, LabVIEW SignalExpress executes both steps simultaneously when you run the project. The steps have a trigger dependency.

If a project contains a DAQmx Generate step and a Create Analog Signal step, and the DAQmx Generate step is generating the signal the Create Analog signal step creates, LabVIEW SignalExpress executes the Create Analog Signal step before the DAQmx Generate step because the steps have a data dependency. The generate step cannot execute until the Create Analog Signal steps passes it an output signal.

You can use trigger sharing or the configuration options on the Step Setup tab of a step to configure execution dependencies for common hardware systems, such as a stimulus/response system.

Acquiring and Generating Signals with Instruments

You can acquire and generate signals using stand-alone instruments or National Instruments modular instruments. Refer to the documentation for these steps for more information about acquiring and generating signals with instruments.

Using Step Execution Dependencies in a Stimulus/Response System

If you want to create a project that measures the response of a system to a stimulus, use a DAQmx Generate step to stimulate the system and a DAQmx Acquire step to acquire the response of the system to the stimulus. To get a valid measurement, execute the DAQmx Acquire step at the same time or after the DAQmx Generate step starts the generation.

If you want to start the generation and the acquisition at the same time, you can establish a trigger dependency between the DAQmx Generate and the DAQmx Acquire steps by sharing a trigger between the steps.

If you want to start the generation before the acquisition, place a checkmark in the **Start this step after** checkbox on the **Execution control** page of the Step Setup tab of the acquisition step. When you run the project, the acquisition step does not execute until after the generation step begins executing. If you want the acquisition step to wait on a step other than the previous generation step, select the step you want to wait for in the **Step to wait for** pull-down menu.

If you want to account for settling times between the execution of dependent steps, you can use the **pre-execution delay** field on the **Execution control** page of the **Step Setup** tab to specify a minimum amount of time to wait between the execution of the steps.

Circular Execution Dependency Errors

When you synchronize multiple devices together, LabVIEW SignalExpress starts the devices in a specific order to ensure correct synchronization.

For example, if you configure a device to start on a digital start trigger and select the start trigger signal of another device as the trigger source, LabVIEW SignalExpress starts the device receiving the trigger before starting the device sending the trigger to ensure that the receiver is ready for the trigger before it is sent.

Similarly, if you configure a step to start after another step by placing a checkmark in the **Start this step after** checkbox and specifying a step in the **Step to wait for** option located on the **Execution Control** tab of the DAQmx Acquire and DAQmx Generate steps, LabVIEW SignalExpress starts the steps based on the order you specify in the **Step to wait for** option.

If you configure your steps to start in the wrong order, you can cause a circular dependency error. The following examples illustrate how you can cause a circular dependency error:

- You configure two steps to start after one another, or you configure two steps to share triggers with each other.
- You configure a step to start on a digital start trigger with the start trigger signal of another step as the trigger source, and you configure the step to start after the step that produces the trigger.

To resolve the circular execution dependency error, modify the order of one or more steps. LabVIEW SignalExpress lists the steps responsible for the error and lists the options you can change to resolve the error.

Note Some steps contain a **Sample clock source** option used when you configure the device to use an external clock. LabVIEW SignalExpress uses the value in this option to ensure the device generating the clock signal generates the signal before LabVIEW SignalExpress programs the device to receive the signal. The **Sample clock source** option also can be responsible for circular execution dependency errors.

Acquiring Data with the DAQmx Acquire Step

With the DAQmx Acquire step, you can acquire analog signals using any NI-DAQmx compatible device in your system.

Configuring the DAQmx Acquire Step

From the step configuration, you can access the device configuration parameters, such as the input channels, the sample rate at which to acquire data, and the number of samples to acquire at each iteration. You also can configure more advanced parameters, such as the start trigger, the reference trigger, and clock. Finally, you can synchronize the DAQmx Acquire step with other hardware steps by configuring a digital trigger and using a trigger signal from another device as the trigger source. You also can synchronize multiple devices by sharing clock and master timebase signals.

Acquisition Timing

To acquire signals with the DAQmx Acquire step, you must understand the relationship between the acquisition timing modes and the LabVIEW SignalExpress run modes. The following table shows how the N Samples, 1 Sample (On Demand), and Continuous Samples acquisition modes work in relation to the Run Once and Run Continuously modes.

	1 Sample (On Demand)	**N Samples**	**Continuous Samples**
Run Once	The device acquires one sample and stops.	The device acquires N samples and stops. The device generates a start trigger at the beginning of the acquisition.	The device acquires N samples and stops. The device generates a start trigger at the beginning of the acquisition.
Run Continuously	The device acquires one sample repeatedly until you stop the execution.	The device acquires N samples repeatedly until you stop the execution. The device generates a start trigger at each iteration.	The device acquires time contiguous blocks of N samples until you stop the execution. The device generates a start trigger at the beginning of the first iteration.

Performance Considerations

When you configure the DAQmx Acquire step in continuous mode, LabVIEW SignalExpress must sustain a minimum loop rate to ensure the continuity of the acquired data. The loop rate is equal to the number of samples to acquire divided by the sample rate. That is, the software must execute fast enough to acquire the signal continuously without losing any data between each buffer transfer. If you display the acquired data on the data viewer or leave any configuration views open, you can reduce the execution speed of LabVIEW SignalExpress and possibly lose data. If you lose data between buffer transfers, the device returns an overwrite error and breaks the continuity of the acquired data.

If the device returns overwrite errors, you can reduce the sample rate or increase the number of points in the buffer, close any configuration views, remove the acquired signal from the Data View, or de-select the **Update Signal Views while Running** option in the View menu.

Note Refer to the *NI-DAQmx Help* for more information about data acquisition.

Generating Data with an NI-DAQmx Device

You can generate any time domain signal in the LabVIEW SignalExpress project with any NI-DAQmx-supported analog output device installed on your system using the DAQmx Generate step. Before you can generate a signal using a DAQmx device, you must first create or import the signal into LabVIEW SignalExpress and then pass the signal to the DAQmx Generate step. You can create the time domain signal using the Create Analog Signal step or you can import it into your project using a snapshot, data log, or the Load from ASCII, Load from LVM, or Load from SPICE steps.

Configuring the DAQmx Generate Step

From the DAQmx Generate configuration view, you can access most of the device configuration parameters, such as the output channel, the output range, and the update rate. By default, the update rate at which the device is configured matches the sample rate of the input signal defined in the Create Analog Signal, Load from ASCII, Load from LVM, or Load from SPICE steps. You can overwrite the update rate with any value by removing the checkmark from the **WT** checkbox and entering a value for **Rate (Hz)**. You can synchronize multiple boards to use the same trigger signal by setting up a digital start trigger and specifying the trigger signal of another device as the trigger source. You also can share clock and master timebase signals to synchronize multiple devices.

Generate Continuously Versus Generate N Samples

To successfully generate signals with the DAQmx Generate step, you must understand the relationship between the generation timing modes and the LabVIEW SignalExpress run modes. The following table describes how the Generate Continuously and Generate N samples timing modes work in relation to the Run Once and Run Continuously modes.

	Continuous Samples	**N Samples**	**1 Sample (on Demand)**
Run Once	The device starts generating the signal continuously and stops. There is no guarantee that the entire signal is generated. National Instruments does not recommend you use the Generate continuously timing mode with the Run Once run mode. The device generates a start trigger at the beginning of the generation.	The device generates the signal once and stops. The device generates a start trigger at the beginning of the generation.	The device generates the signal once and stops.
Run Continuously	The device generates the signal continuously until the user stops the execution. If the input signal changes while the project is running, this step reloads the input signal into the output buffer of the device without	The device generates the input signal discontinuously until you stop the execution. If the input signal changes while the project is running, this step reloads	The device generates the input signal continuously until you stop the execution. If the input signal changes while the project is

	stopping the device. The device only generates one start trigger at the beginning of the generation.	the input signal into the output buffer of the device. The device generates a start trigger at each iteration.	running, this step reloads the input signal into the output buffer of the device.

Generating a Non-repetitive Noise Signal or Phase Continuous Signal

NI-DAQmx-supported devices can update their output buffer while running and generate phase continuous or non-repetitive noise signals.

Complete the following steps to generate a non-repetitive noise signal.

1. Place the Create Analog Signal step in the Project View.
2. In the **Signal type** pull-down menu, select **Noise Signal**.
3. Enter 10k in the **Sample rate** field.
4. Make sure there is no checkmark in the **Repeated signal** checkbox.
5. Click the **Add Step** button and select **Generate Signals»DAQmx Generate»Analog Output** and select **Voltage** or **Current**.
6. On the **Configuration** page of the Step Setup tab for the DAQmx Generate step, select **Continuous Samples** from the **Generation Mode** pull-down menu.
7. Run LabVIEW SignalExpress continuously. The Create Analog Signal step produces a new noise pattern at each iteration and passes the pattern to the DAQmx Generate step that loads it into the analog output device, resulting in a non-repetitive noise generation.

You can use the Create Analog Signal and the DAQmx Generate steps to generate a phase continuous signal of any arbitrary frequency. Complete the following steps to generate a phase continuous noise signal.

1. Place the Create Analog Signal step in the Project View.
2. In the **Signal type** pull-down menu, select a Sine Wave and specify an arbitrary **Frequency**.
3. Remove the checkmark from the **Repeated signal** checkbox so that each iteration of the signal is contiguous to the previous iteration.
4. Run LabVIEW SignalExpress continuously. The Create Analog Signal step produces a continuous signal with increasing timestamp and phase continuity.

To maintain phase continuity of a generated signal, LabVIEW

SignalExpress must sustain a minimum loop rate. The minimum loop rate is equal to the sample rate divided by the number of samples in the signal. If LabVIEW SignalExpress cannot sustain the minimum loop rate, remove any unnecessary graph display plots in the Data View and close any configuration views. If the software still cannot sustain the minimum loop rate, decrease the sample rate or increase the buffer size to decrease the loop rate.

Refer to the Continuous Noise Generation and Finite Acquisition (Non Regeneration) example located in the SignalExpress\examples\DAQmx directory for an example of generating a true noise signal.

Generating a Repetitive Signal with an NI-DAQmx Device

You can use LabVIEW SignalExpress to generate a repetitive signal continuously. With a repetitive signal, the signal is created and loaded into the device only once, which means that there are no minimum loop rates required. You can perform this type of repetitive signal generation with a much higher sample rate than a non-repetitive noise or phase continuous signal generation. When generating a repetitive signal, you can generate using the maximum sample rate the device allows.

Complete the following steps to perform a repetitive signal generation.

1. Create the repetitive signal by importing the signal from a file using the Load from ASCII, Load from LVM, or Load from SPICE steps. You also can use the Create Analog Signal step. If you use the Create Analog Signal step, place a checkmark in the **Repeated signal** checkbox.
2. Pass the signal to the DAQmx Generate step, which loads data to the output buffer of the device only when the data changes. Because the file import steps and the Create Analog Signal step produce their output signal only at the first iteration, the DAQmx Generate step loads the signal into the device at the first iteration and repetitively generates the signal until you press the **Stop** button.

Refer to the Continuous Tone Generation and Finite Acquisition (Regeneration) example located in the SignalExpress\Examples\DAQmx directory for an example of generating a repetitive signal.

Generating a DC Signal with NI-DAQmx Devices

In the DAQmx Generate step, select **1 Sample (On Demand)** in the Generation Mode pull-down menu. You can select a programmatic input to generate, or you can remove the checkmark from the **Use Programmatic Input** checkbox and specify a value to generate in the **Value to Write** field.

Note Refer to the *NI-DAQmx Help* for more information about generating data.

Master Timebase and Reference Clock Sharing

Certain types of NI-DAQmx-supported devices support master timebase or reference clock sharing between multiple devices, which is similar to trigger and sample clock sharing. For example, E series devices can share master timebases, and M series devices can share reference clock signals.

Timebase settings appear on the **Advanced Timing** page of the Step Setup tab for DAQmx Acquire and DAQmx Generate steps. To share a master timebase or reference clock between two devices, navigate to the **Step Setup** tab for the DAQmx Acquire or DAQmx Generate step for one of the devices and select **External** from the **Master Timebase** or **Reference Clock** pull-down menu. Select the device you want to use as the source from the **Master Timebase Source** or **Reference Clock Source** pull-down menu. If you are sharing a timebase or reference clock among more than two devices, be sure to always pick the same device as the timebase or clock source.

Displays

A project can contain multiple Data View tabs, and one **Data View** tab can contain multiple displays. A display is a section of a **Data View** tab on which you can view signals in the project. Depending on the data type of the signal you add to a display, you can select from multiple display formats, such as graphs, charts, and various numeric representations.

LabVIEW SignalExpress displays signals of the same type in the same display by default, but you can add additional displays by clicking the **Add Display** button or by right-clicking an existing display and selecting one of the **Data View»Add Display** options from the shortcut menu. To remove a display, right-click the display and select **Data View»Remove Display** from the shortcut menu.

Adding and Removing Signals

To add signals to a display, drag a signal from the Project View to the display or right-click the display, select **Signals»Add Signal** from the shortcut menu, and select the signal you want to add. To remove signals from a display, right-click the display, select **Signals»Remove Signal** from the shortcut menu, and select the signal you want to remove. You also can right-click a signal in the Project View and select **Display»Existing Data View** or **Display»New Data View** to add the signal to a new display on the active **Data View** tab or to a new display on a new **Data View** tab.

Configuring Display Properties

You can use the Display Properties dialog box to configure properties for all types of displays. The display properties you can configure depend on the format of the display and can include titles, format and precision, plots, scales, cursors, and signal order. Click the **Properties** button or right-click a display on the **Data View** tab and select **Properties** from the shortcut menu to display the **Display Properties** dialog box.

Additional Display Options

You can use the shortcut menu for a graph display to display a preview graph, a graph legend, cursors, an event viewer, or alarms. Right-click a graph and select **Visible Items** from the shortcut menu to select these options.

Display Formats

You can select from multiple format options for displaying data on the Data View tab. To change the format of a display, right-click the display and select one of the **View As** options from the shortcut menu. The number of options available depends on the data type of the signal.

Note The **View As** option only appears in the shortcut menu of a display if the display contains data that you can view in a different display format.

Graphs

Graphs are the most common type of display in LabVIEW SignalExpress. Time-domain, frequency-domain, and xy signals always appear on graphs on the **Data View** tab. One graph can display multiple signals only if the signals are of the same type. For example, if you try to add a frequency-domain signal to a graph that displays a time-domain signal, a new graph appears to plot the frequency-domain signal because the signals are different signal types.

Additional Display Format Options

In addition to graphs, you can choose from the following formats to display data on the **Data View** tab:

Display Format	Supported Data Types	Description
Numeric Display	Scalar (integers only)	Displays the numeric value of an integer scalar.
Vertical LED	Boolean	Displays an LED that indicates the value of a Boolean signal.
Chart Display	Boolean, Scalar	Displays a plot of a Boolean or scalar signal and a legend that includes the name and current value of the signal. The chart display updates periodically and maintains a history of the signal data previously stored. Right-click the plot and select **History Length** from the shortcut menu to specify the number of

		samples to store in a chart history. The number of samples is the number of data points LabVIEW SignalExpress uses to plot the signal. You also can specify how a chart display updates the plotted data by right-clicking the plot and selecting one of the following **Update Mode** options from the shortcut menu: • **Strip Chart**—Shows data continuously scrolling from left to right. • **Scope Chart**—Shows one item of data scrolling partway across the plot from left to right. When the data reaches the right border, LabVIEW SignalExpress erases the plot and begins plotting again from the left border. • **Sweep Chart**—Similar to a scope chart except that when LabVIEW SignalExpress begins plotting again from the left border the plot shows both the old data and the new data separated by a vertical line.
Slider	Scalar	Displays a vertical slide filled to the value of the signal.
Table Display	Boolean, Scalar	Displays a table that includes the name and current value of the signal.
Tank	Scalar	Displays a tank filled to the value of the signal.
Thermometer	Scalar	Displays a thermometer filled to the value of the signal.
Gauge	Scalar	Displays a gauge with a needle that indicates the value of the signal.
Large Display	Boolean, Scalar	Displays the value and units of the signal in large text.

	(double-precision only), String	
Meter	Scalar	Displays a meter with a needle that indicates the value of the signal.

Viewing and Analyzing a Logged Signal

After you log a signal, you can switch to the Playback work area to play back and analyze the logged signal. The Playback work area is similar to the Monitor/Record work area except that the Data View tab appears with a time bar and various buttons you can use to navigate a logged signal, and you can use logged signals as step inputs. Complete the following steps to view a logged signal in the Playback work area and run the logged signal through a Processing or Analysis step.

1. In the Monitor/Record work area, log a signal.
2. Select **Playback** or the name of the log you want to play back from the pull-down menu that appears above the Project View to switch to the Playback work area. You also can select **View»Work Areas»Playback** to switch to the Playback work area.

Note If you select a specific log from the pull-down menu above the Project View, LabVIEW SignalExpress makes that log the active log when you switch to the Playback work area.

2. Drag a logged signal from the Logged Data window to the **Data View** tab.
3. Click the **Run** button. LabVIEW SignalExpress plays back the logged signal at the speed at which you recorded the signal.
4. (Optional) Use the buttons and the pull-down menu on the time bar to adjust how LabVIEW SignalExpress plays back the logged signal. You can change the playback speed, play back the signal repeatedly, play back the signal from a specific point in time, or run one iteration (block) of the logged signal at a time. You also can move the pointer on the time bar to navigate through the logged signal manually.
5. Add a step that requires an input signal, such as the Filter step, to the Project View. LabVIEW SignalExpress automatically selects the active log as the step input.
6. Drag the output of the step to the **Data View** tab and click the **Run** button. LabVIEW SignalExpress runs the logged signal through the step block by block and displays the results on the **Data View** tab.

You can use the Playback Options tab to configure advanced playback options. For example, you can use the **Playback Options** tab to select subsets of logged signals to play back and analyze, change the block size of logged signals, and to configure overlap of blocks of signal data.

Changing Plot Order on Graphs

When you add a signal to a graph, LabVIEW SignalExpress assigns the signal a color and a position in the plot order. Select **Visible Items»Legend** to display the graph legend and see the plot order of the signals. By default, the last signal you add to the graph is the last signal in the plot order in the legend. The order in which you add the signal dictates the color LabVIEW SignalExpress assigns to the plot.

Complete the following steps to change the plot order on a graph.

1. Right-click a graph display and select **Properties** from the shortcut menu to display the Display Properties dialog box.
2. On the Signal Order page, click the **Move Forward** button to move the signal up one spot in the plot order. Click the **Move Backward** button to move the signal down one spot in the plot order. Click the **Move To Front** button to move the signal to the top of the plot order. Click the **Move To Back** button to move the signal to the end of the plot order.
3. Click the **OK** button to close the dialog box and apply the changes.

When you move signals in the plot order, the signals change color so that the first signal is always white, the second signal is always red, etc. You can change the color of a signal by clicking the color next to the signal in the graph legend and selecting a color from the palette that appears.

Aligning Signals

Signals from different sources, such as different hardware devices or files, vary in amplitude and timing and might not align on a graph in a useful way for comparing the signals. You can align signals by configuring time stamps on a graph, or you can use the Interactive Alignment step.

Aligning Signals Using the Interactive Alignment Step

Complete the following steps to align two time-domain signals on a graph using the Interactive Alignment step.

1. Add the Interactive Alignment step to the Project View. The step selects the two most recent signals in the project as inputs, and displays them on the **Input Signals** graph in the Configuration View.

   Note The Interactive Alignment step can operate only on time-domain waveform signals.

2. In the Configuration View, configure the comparison using one of the following methods:
   - Configure the comparison manually. In the **Input Signals** graph, select the test signal, plotted in red, and drag it to the position you want in the graph for the comparison. You can expand the test plot manually in both directions by holding the <Alt> key while you drag the plot. The step computes the **x-offset**, **y-offset**, **x-gain**, and **y-gain** as you move the plot.
   - Configure the comparison based on a built-in algorithm. Select one of the auto modes from the **Mode** pull-down menu.
3. Switch to the Data View and drag the **comparison signal** output to a view to see the results of the comparison.

Cursors

When you display signals in graph or chart displays on the Data View tab, you can use cursors to determine specific values within the signals. Right-click a graph and select **Visible Items»Cursors** from the shortcut menu to add cursors to the graph and display a cursor legend below the graph. The cursor legend displays the x- and y-values of the signal at the point where each cursor intersects the signal, the slope between the x- and y-values, and the inverse and derivative values of the slope.

Cursors appear as perpendicular lines that intersect at a point on the plot of a signal. When you add cursors to a graph, LabVIEW SignalExpress always displays two cursors, and the cursors always are linked to a plot on the graph so that the cursor remains aligned with the signal when you move the cursor. If the graph contains multiple signals, LabVIEW SignalExpress links the cursors to the plot of the first signal in the signal order by default.

Changing Cursor Position

When you add cursors to a graph, LabVIEW SignalExpress enables the **Cursor Movement Tool** on the graph palette automatically. With the **Cursor Movement Tool** enabled, you can drag cursors to different positions on the plot of a signal. You also can right-click the graph and select ***Cursor Name*»Set Position** to move the vertical cross hair of the cursor to the location where you right-click the graph.

Finding Peak Values

You can use cursors to find peak values of a signal. Right-click the graph and select ***Cursor Name*»Next Peak** or ***Cursor Name*»Previous Peak** to move the cursor to the corresponding peak value. Use the Cursors page of the Display Properties dialog box to define the peak threshold and peak width that a cursor uses to recognize peak values.

Changing Linked Plots

If a graph contains multiple signals, you can change the signal plot to which a cursor is linked. To change the linked plot of a cursor, click the signal name in the cursor legend and select a signal from the shortcut menu, right-click a cursor in the cursor legend and select **Link To Plot»*Plot name*** from the shortcut menu, or right-click the graph and select ***Cursor Name*»Link To Plot»*Plot name*** from the shortcut menu. You also can use the **Linked to plot** pull-down menu on the **Cursors** page of the **Display Properties** dialog box to change the linked plot of a cursor.

Customizing Cursors

After you add cursors to a graph, you can use the **Cursors** page of the **Display Properties** dialog box to customize the style, appearance, orientation, and behavior of the cursors. You also can use this page to specify values a signal must cross before LabVIEW SignalExpress recognizes peak values, export measurement values from the cursor legend to use as step inputs, and bind exported cursor measurements to step parameters so the parameter to which you bind a cursor measurement updates to use the value of the measurement.

To display the **Cursors** page, right-click a cursor in the cursor legend and select **Cursor Properties** from the shortcut menu or right-click the graph and select ***Cursor Name*»Properties** from the shortcut menu.

Note You also can click the **Properties** button on the **Data View** tab or right-click a display and select **Properties** from the shortcut menu to display the **Display Properties** dialog box, then navigate to the **Cursors** page.

Binding Cursor Measurement Values to Step Parameters

When you display cursors on a graph of a signal, you can export the measurement values that appear in the cursor legend to use as step inputs, and you can bind those values to step parameters. Binding cursor measurements to step parameters can be useful when you want to use a very specific value for the parameter. For example, you might want to use the Subset and Resample step to create a subset of a signal. If you want the subset to begin at a specific point in the input signal, you can bind the x-value of a cursor on the input signal to the **Start position** parameter of the step.

Complete the following steps to bind a cursor legend value to a step parameter.

1. Add a signal to a graph display on the Data View tab, right-click the display, and select **Visible Items»Cursors** from the shortcut menu to display cursors and the cursor legend.
2. Right-click a cursor in the cursor legend and select **Cursor Properties** from the shortcut menu to display the Cursors page of the Display Options dialog box.
3. On the **Cursors** page, select the **Measurements** tab.
4. In the **Cursor Measurements** table, find the measurement you want to bind to the step parameter and click the button that appears in the **Bind** column to display the **Binding Selection** dialog box. This dialog box displays all the step parameters to which you can bind cursor measurements.
5. Place a checkmark in the checkbox next to the parameter(s) to which you want to bind the cursor measurement.
6. Click the **OK** button to close the **Binding Selection** dialog box and bind the cursor measurement to the parameter. LabVIEW SignalExpress places a checkmark in the **Export** checkbox for the measurement in the **Cursor Measurements** table.

Note When you bind a cursor measurement to a step parameter, LabVIEW SignalExpress exports the cursor measurement automatically.

7. Click the **OK** button to close the **Display Properties** dialog box.

To undo binding a cursor measurement to a step parameter, follow the preceding steps until you display the **Binding Selection** dialog box. Remove the checkmark from the checkbox next to a parameter and click the **OK** button to undo binding the cursor measurement to the parameter.

Data Grouping

You can analyze multiple data channels at the same time. Drag a group of channels from the Project View into the Data View to view all signals from the group on the same display. You also can send groups of data through analysis steps. When selecting an input signal for a step, select the **All Elements** option. The step outputs results for every channel within the data group.

Creating Subgroups

You can create a user-defined subgroup of data. Right-click a step output and select **Define Subgroups** to display the **Define Subgroups** dialog box. Click the **Create a new subgroup** button to create a new subgroup and specify a name for the subgroup. Select a signal from the **Signals** column and click the **Copy signals into the selected subgroup** button to add the selected signals into the selected subgroup.

Managing Hardware with the Channel View

The Channel View is a central location for viewing and configuring hardware and shared variables. You can use the Channel View to manage and configure hardware devices and channels and to acquire signals in LabVIEW SignalExpress.

In the **Physical Channel** column of the Channel View, click the plus sign next to a device name to view the physical channels of the device. Each channel and device appear with a checkbox in the **Acquire** column. Use the **Acquire** checkbox to specify whether to acquire data from the channel or device. When you place a checkmark in the **Acquire** checkbox, additional columns of measurement configuration options appear in the Channel View, and LabVIEW SignalExpress automatically places a data acquisition step in the Project View. LabVIEW SignalExpress configures the step to match the options you specify in the Channel View.

Note You also can use the Properties window to configure channels or devices.

Managing Shared Variables with the Channel View

The Channel View is a central location for viewing and configuring hardware and shared variables. You can use the Channel View to view and read the values of shared variables on the local machine or local network.

The Channel View displays hardware devices by default. Select **Shared Variables** from the **View** pull-down menu at the top of the Channel View to display all the shared variables that exist on the local machine or local network. In the **Shared Variable Name** column, expand **Localhost** to display shared variables that are available on the local machine. Expand **Network Neighborhood** to display machines on the local network. You also can click the **Add Machine** button and enter a machine name or IP address to add a machine to the Channel View.

Networks, machines, and shared variables appear with checkboxes in an **Acquire** column. Use the **Acquire** checkbox to specify whether to read the values of shared variables. When you place a checkmark in the **Acquire** checkbox, a **Sample Period (s)** column appears, and LabVIEW SignalExpress automatically places a Read Shared Variables step in the Project View. LabVIEW SignalExpress configures the step to match the options you specify in the Channel View.

Note You also can use the Properties window to quickly configure multiple shared variables.

Place a checkmark in the **Acquire** checkbox for a machine to read the values of all the available shared variables on that machine. If you select multiple shared variables, LabVIEW SignalExpress uses one Read Shared Variables step to read the values of any shared variables that have the same sample period.

Importing and Exporting Channel View Data

You can export data from the Channel View to a Microsoft Excel spreadsheet and import the data back into the same project or into a new project. You can use this feature to share Channel View configurations among multiple projects or to edit configuration settings from Microsoft Excel, which can be useful when the Channel View contains a large number of items. Select **File»Export»Channel View to Excel** or right-click in the Channel View and select **Export To»Microsoft Excel** to export data from the Channel View to Microsoft Excel.

Note If you intend to import data from the Microsoft Excel spreadsheet back into LabVIEW SignalExpress, you must not edit the top-level header or any of the column headers in the spreadsheet. LabVIEW SignalExpress uses the headers to populate the Channel View correctly. You can edit any other values in the spreadsheet.

Select **File»Import»Channel View from Excel** or right-click in the Channel View and select **Import From»Microsoft Excel** to import Channel View data from a Microsoft Excel spreadsheet. When you import data from a Microsoft Excel spreadsheet, an **Importing Excel Data** window appears and displays a preview of the Channel View. Any values that LabVIEW SignalExpress cannot import successfully appear with error icons in the **Importing Excel Data** window. Move the cursor over an error icon to display more information about the error.

Importing Signals from Files

Use the steps in the Load/Save Signals menu to import files into LabVIEW SignalExpress. You can import signals from ASCII text, .lvm, SPICE, PSpice, and Multisim files. After you import data, you can use the data as an input signal for other steps.

Saving Data to Files

You can save or export a signal or value to a file in LabVIEW SignalExpress in the following ways:

- Right-click an output on a step and select **Save Value** from the shortcut menu to save the signal or value to a text file.
- Right-click an output on a step and select **Copy Value** from the shortcut menu to copy the data to the clipboard. You can paste the data into other applications such as Microsoft Excel and Notepad.
- Open Microsoft Excel and drag the step output to an Excel worksheet.
- Use the Save to ASCII/LVM step to write signals or values to a text file after a step or series of steps completes execution.
- Use LabVIEW SignalExpress data logging features to create logs of your signals in the .tdms file format.
- Right-click a logged signal in the Logged Data window and select the **Convert to ASCII** option to save logged data in the ASCII format.

When you use the Save to ASCII/LVM step, LabVIEW SignalExpress saves signals and values in the LabVIEW measurement data file format, which is a tab-delimited text file you can open with a spreadsheet application or a text-editing application. Refer to the Specification for the LabVIEW Measurement File (.lvm) application note for more information about this file format.

Snapshots

Snapshots allow you to save a record of the current values of any signal in your project. You can view snapshot values on the Data View tab. You also can save these values to file or run them through analysis steps. You can use snapshots to compare data within the same project. For example, you can acquire data and create a snapshot of the data when the acquisition task is complete. Complete a new acquisition task with different settings and create another snapshot of the data after you complete the acquisition task. Drag each of the snapshots to the **Data View** tab to view and compare the data.

Select **Operate»Create Snapshot** to open the **Create Snapshot** dialog box. Use this dialog box to select the signals to include in the snapshot, and click the **OK** button to close the dialog box and create a snapshot of the signals you select. The snapshot appears in the Logged Data window. In the **Logged Data** window, expand the snapshot and right-click a signal in the snapshot to display a shortcut menu with various options, such as saving the snapshot of that signal to a text file and/or sending the signal snapshot data to an analysis step.

LabVIEW SignalExpress automatically saves snapshot data in the project file. You also can import snapshots from another project. Select **File»Import»Snapshots From Another Project** to select the project file from which you want to import snapshots.

Defining Data Values

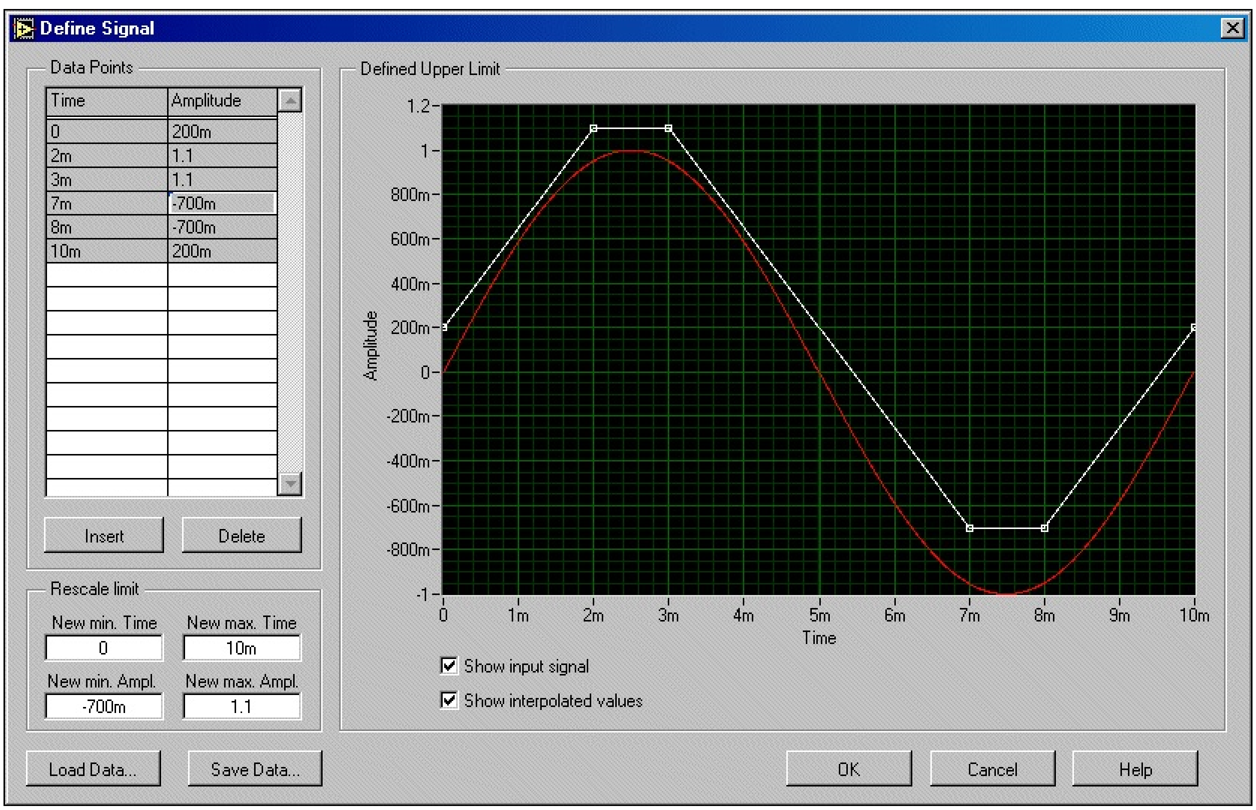
You can use the Define Signal dialog box to define a data value for a limit test.

To define a corner Data Point, click the **Insert** button below the table or enter the X and Y values in the table. The new point appears on the graph. You can change the position of a point by clicking the point on the graph and moving it. If you move any point past another data point, the table automatically reorders and increases the X column values.

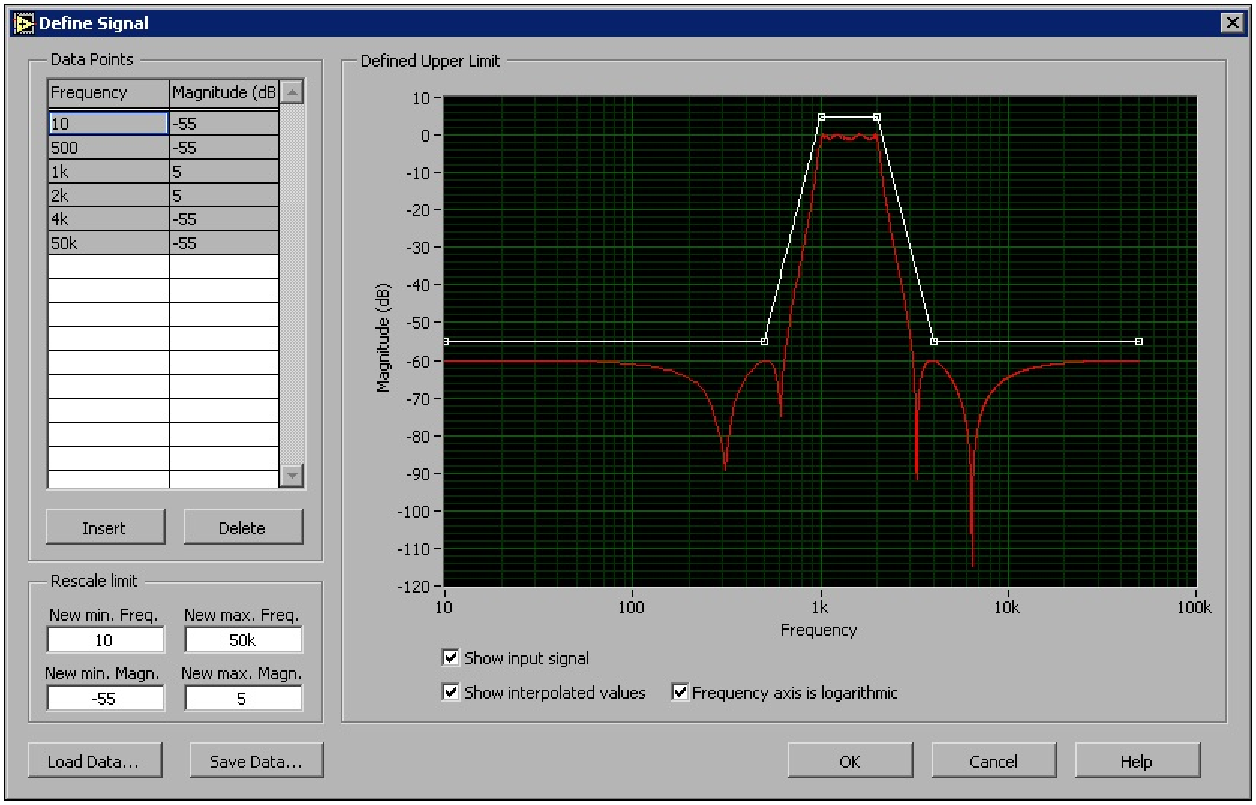
You can rescale the entire limit signal by entering new values in the **New min.** and **New max.** fields in the **Rescale limit** section.

You can save the data points that define the limit to a file using the **Save Data** button. You also can load data points from a file using the **Load Data** button. LabVIEW SignalExpress uses the LabVIEW measurement .lvm format.

The following graphic is an example of using the **Define Signal** dialog box to define the upper limit of a time-domain signal.



The following graphic is an example of using the **Define Signal** dialog box to define the upper limit of a frequency-domain signal using logarithmic frequency axis.



Timestamps in LabVIEW SignalExpress

Time-domain signals use a timestamp, which is a value that provides information about when the signal began. Depending on the source of the signal and the actual processing steps you added to the project, the timestamp of the signal can be an absolute or relative timestamp, or LabVIEW SignalExpress can ignore the timestamp.

An absolute timestamp value represents an actual date and time, such as 12:37 p.m., April 6. Some data acquisition boards, such as the E-Series acquisition board, can return absolute timestamps, which you can use for accurate datalogging.

A relative timestamp represents a time value that is relative to a known reference event, such as the start of an acquisition or the trigger point. For example, a continuous acquisition of a signal might have a timestamp value relative to the trigger point of the signal. Timestamps also can be relative to other timestamps. For example, you can extract a subset of a signal starting at a time that is relative to the beginning of that signal.

If you ignore the timestamp, you can perform operations like addition or subtraction on the two signals independently of the actual timestamp values. For example, you can add the signals as if they were aligned perfectly, which is useful if you want to compare a triggered signal to a model that originates from a file.

Several processing steps work on continuous signals and assume that the data are contiguous. For example, the Filter step resets the filter history when you call it the first time, but then in subsequent calls, the step filters the incoming time signal segments using the history information of the previous segment as long as the timestamps are contiguous. If the Filter step detects a time discontinuity, such as a missing segment or an inconsistency in timestamps, the Filter step resets itself and returns a warning. Using relative timestamps in a continuous acquisition does not conflict with the timestamp continuity requirements.

National Instruments recommends using relative timestamps as the default mode for all steps that can deliver a continuous signal, such as steps that acquire or create signals. Use absolute timestamps only when you require absolute date and clock information. Avoid ignoring timestamps when possible.

Shared Variables

Shared variables are application-independent configured software items that enable you to send data across projects or across a network.

Viewing Shared Variables

You can use the Channel View to display and manage shared variables that exist on the local machine or local network. In the Channel View, select **Shared Variables** from the **View** pull-down menu. Expand **Localhost** to display shared variables that are available on the local machine. Expand **Network Neighborhood** to display machines on the local network. LabVIEW SignalExpress recognizes shared variables created in LabVIEW SignalExpress and LabVIEW, as well as data published using DataSocket technology or data that meets OPC specifications.

Note In this help topic, the term *shared variable* also applies to data published using DataSocket technology or data that meets OPC specifications. Refer to ni.com for more information about DataSocket technology. Refer to the OPC Foundation Web site for more information about OPC specifications.

Reading Shared Variables

Use the Read Shared Variables step to read shared variables in LabVIEW SignalExpress. The Read Shared Variables step reads the value of a shared variable and returns the value as an output you can send to other steps. You also can use the Channel View to read the value of a shared variable. Place a checkmark in the **Acquire** checkbox for a shared variable to read the value of the variable. LabVIEW SignalExpress automatically configures a Read Shared Variables step and adds the step to the Project View. Place a checkmark in the **Acquire** checkbox for a machine to read the values of all the available shared variables on the machine. LabVIEW SignalExpress uses one Read Shared Variables step to read the values of all the shared variables.

Note You can use the **Sample Period (s)** column of the Channel View or the **Sample period (s)** parameter of the Read Shared Variables step to specify how frequently LabVIEW SignalExpress reads the value of the shared variable.

Writing to Shared Variables

You can write signals in a LabVIEW SignalExpress project to shared variables so that other projects can use the value of the signal. Right-click a step input or output signal and select **Write to Shared Variable** from the shortcut menu to write the value of the signal to a shared variable. The name of the signal appears with a globe icon ( ) that indicates it is a shared variable. You also can right-click a shared variable input or output and select **Write to Shared Variable** from the shortcut menu to stop sharing the value of the signal.

LabVIEW SignalExpress automatically shares any shared variables you create on the local network. The shared variable appears at the path *hostname*\NI_SIGX_*projectname*, where *hostname* is the name of the machine on which you create the variable and *projectname* is the name of the LabVIEW SignalExpress project in which you create the variable.

Note LabVIEW SignalExpress removes shared variables from the local network when you close the project that creates the shared variable(s).

Acquire

Specifies whether to acquire, or read the value of, the shared variable. LabVIEW SignalExpress uses the Read Shared Variables step to acquire the value of the shared variable.

Name

Displays the name of the shared variable.

Path

Displays the path to the shared variable.

Sample Period (s)

Specifies the period (in seconds) at which to read data from the shared variable.

Controlling the Execution of a Project

When you execute, or run, a LabVIEW SignalExpress project, LabVIEW SignalExpress executes all the steps that the project contains at least once. You can control the execution of a project by ordering steps in the Project View to create execution dependencies, configuring the run mode of a project, using Execution Control steps, or using hardware.

Controlling Project Execution by Ordering Steps

The order in which steps execute partially depends on the order in which they appear in the Project View. Steps that inherit signals from other steps execute sequentially. For example, steps that provide output signals must appear in the Project View before steps that require those signals as inputs. A step that receives a signal has an execution dependency on the step that sends the signal. LabVIEW SignalExpress draws blue execution loops around groups of steps with mutual execution dependencies.

Controlling Project Execution by Configuring Run Modes

You can configure how many times an entire project runs by configuring the run mode for the project. You can configure LabVIEW SignalExpress to run a project once, for a number of iterations you specify, or for an amount of time you specify.

Controlling Execution with Steps

You can use the following Execution Control steps to control the execution of some or all steps in a project:

- Sweep—Iterates a set of steps the number of times you specify.
- Conditional Repeat—Iterates a set of steps until a specified condition occurs.
- Trigger—Sets a trigger condition and returns a section of a continuous signal when the condition occurs.
- Sequence—Pauses and resumes execution of steps in a project based on the configuration you specify. The Sequence step can pause the execution of a step without stopping the execution of the entire project. Because the Sequence step can pause the execution of other steps, you can use the Sequence step to allow multiple steps in the same project to use the same hardware.

Controlling Execution with Hardware

You can use the triggering functionality of various instruments to control the execution of steps or a project. Use the Acquire Signals or Generate Signals steps to configure triggers when you use hardware with LabVIEW SignalExpress.

Run Modes

By default, LabVIEW SignalExpress runs projects continuously when you click the **Run** button. However, you can configure LabVIEW SignalExpress to run a project once, for a number of iterations you specify, or for an amount of time you specify. If you run a project once, the project is in Run Once mode. If you run a project continuously, the project is in Run Continuous mode. If you run a project for a number of iterations or an amount of time you specify, the project is in Run Continuous mode until the stop condition occurs.

Use Run Once mode if you want the steps in a project to execute only once. For example, if you use a Sweep step to control the execution of steps in a project, you might not want the Sweep step to continue running after it iterates through the values you specify when you configure the Sweep step.

Use Run Continuous mode if a project contains steps that operate on signals iteratively, such as steps that filter or average signals. For example, if you use the Power Spectrum step to average data in a project, the step averages the data over time with each iteration of the project. If you do not run the project continuously, you cannot average the data because LabVIEW SignalExpress resets the Power Spectrum step each time you run the project in Run Once mode.

If you configure a project to run for a specified number of iterations or a specified time in seconds, the project runs in Run Continuous mode until the stop condition occurs. For example, if you configure a project to run for 30 seconds, the project executes as if you set the project to run continuously, clicked the **Run** button, and then clicked the **Stop** button 30 seconds later.

Note If you configure a project to run for a specific amount of time, LabVIEW SignalExpress uses software timing to determine how long the project runs. You cannot use hardware timing to control how long a project runs.

Configuring Run Modes

Click the down arrow on the **Run** button and select **Run Continuously**, **Run Once**, or **Configure Run** to specify the run mode you want to use. You also can access run mode configuration options from the Operate menu.

When you select **Configure Run**, the Configure Run dialog box appears. You can use this dialog box to specify a number of iterations or an amount of time in seconds for which you want the project to run. You also can create a snapshot of all the signals in the project when the project finishes running. After you configure the run mode, the icon on the **Run** button updates to reflect the configuration. You configure run modes on a per-project basis. When you save a project, LabVIEW SignalExpress also saves the configuration of the **Run** button.

Conversions to and from LabVIEW VIs

When you convert a LabVIEW SignalExpress project to a LabVIEW block diagram, the generated VI runs in the mode you configured for the project unless you edit the VI to change how the VI runs. If you import a LabVIEW SignalExpress project from LabVIEW, you can configure the run mode through the pull-down menu on the **Run** button or the **Operate** menu.

Note If a project includes a Run LabVIEW VI step, the run mode you configure does not override the execution of elements of the LabVIEW VI, such as loops.

Resetting a Project

When you run a project, you can reset all steps to their initial state by clicking the **Reset All** button. Resetting restarts the process for the steps. For example, if you have a step that averages a series of numbers, clicking the **Reset All** button restarts the averaging from the beginning.

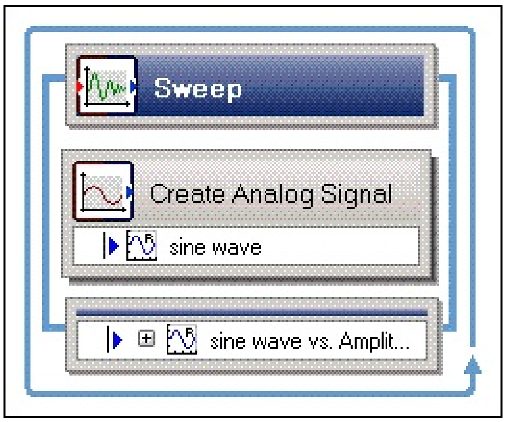
Note Resetting does not reset parameters to their default values.

You also can revert a project to its last saved state by selecting **File»Revert Project.**

Sweeping

The Sweep step controls the execution of other steps in a project. Use the Sweep step to iterate a set of measurement steps the number of times you specify. Each iteration modifies one or more parameters of one or more steps in the measurement. Most steps in LabVIEW SignalExpress have sweepable parameters.

When you add a Sweep step to the Project View, the step appears as a loop. Place steps inside the loop of the Sweep step to control the execution of those steps using the Sweep step. The following illustration shows a Sweep step that is controlling the execution of a Create Analog Signal step.



You can use the Step Setup tab for the Sweep step to configure a range of values to send to the steps the Sweep step controls. You also can define outputs for the Sweep step and configure parallel and nested sweep operations.

Refer to Chapter 5, *Performing Sweep Measurements*, in the Getting Started with LabVIEW SignalExpress manual for more information about sweeping.

Configuring a Sweep Range

The Sweep step controls the execution of other steps in a project. You can use the Sweep step to configure a range of values to send to the steps it controls.

For example, you can use the Sweep step to change the frequency value of a signal you create with the Create Analog Signal step. On the Step Setup tab for the Sweep step, add **Frequency (Hz)** to the **Sweepable parameters** list. Then use the **Configuration** options to select the type of sweep you want to perform and define the range of values you want to use for the sweep. You can define the range as a linear range, an exponential range, a list of points, a formula, or by using a file that contains a list of values. When you run the project, the Sweep step uses the **Configuration** settings you specify to change the frequency value of the Create Analog Signal step with each iteration of the sweep.

When you configure the Sweep step, the Project View updates so that the Sweep step encloses the steps it controls. In the previous example, the Sweep step encloses the Create Analog Signal step. When you run a project with a sweep, the Sweep step generates each value in the sweep range and passes the values to the steps it controls to define new values for each iteration. You also can define the outputs of a sweep operation.

You can use the Sweep step to perform a parallel sweep that sets values on multiple parameters and multiple steps simultaneously. You also can perform a nested sweep by enclosing a Sweep step within another Sweep step.

Defining Sweep Operation Outputs

You can specify outputs of a sweep operation to track how the parameters you sweep affect other values in the project. For example, if you sweep the frequency of a stimulus signal, you might want to track the RMS voltage level during the sweep operation to see how the RMS level changes in response to the changing frequency. You then can view a graph of the RMS level versus frequency on the Data View tab.

Complete the following steps to define outputs for a sweep operation.

1. Configure the Sweep step using the **Sweep Configuration** page of the Step Setup tab for the step. To define a sweep operation output, you must specify at least one parameter in the **Sweepable parameters** field.
2. Switch to the **Sweep Output** page of the **Step Setup** tab for the Sweep step.
3. Click the **Add** button to display the Edit Sweep Output dialog box. The **Output (Y-Axis)** field displays outputs of steps in the sweep operation. The **Input (X-Axis)** field displays the sweepable parameters you specified on the **Sweep Configuration** page.
4. Select an output from the **Output (Y-Axis)** field. When you graph the sweep operation output, this value appears on the Y-axis of the graph. This is the value that you expect to change as a result of the changing value of the parameter you sweep.
5. Select a sweepable parameter from the **Input (X-Axis)** field. When you graph the sweep operation output, this value appears on the X-axis of the graph. This is the parameter that changes value with each iteration of the sweep.
6. Click the **OK** button. The sweep operation output appears as an output of the Sweep step in the Project View.

Note If you specified two scalar values for the **Input (X-Axis)** and **Output (Y-Axis)**, the output is a single XY waveform. The number of data points in the waveform matches the **Number of points** you specify on the **Sweep Configuration** page of the **Step Setup** tab.

If you specified a waveform for the **Input (X-Axis)** or **Output (Y-Axis)**, the output is an expandable group of waveforms. Each waveform in the group is the value of the waveform during an iteration of the sweep. The total number of waveforms matches the **Number of points** you specify on the **Sweep Configuration** page of the **Step Setup** tab.

7. Drag the sweep operation output to the **Data View** tab to plot the output on a graph. For groups, you can plot an element of the group or the entire group.

You also can define advanced sweep operation outputs that compare the outputs of steps affected by the sweep operation instead of comparing one step output and one sweepable parameter.

Defining Advanced Sweep Operation Outputs

Advanced sweep operation outputs compare the outputs of steps affected by a sweep operation instead of comparing one step output and one sweepable parameter. Complete the following steps to define advanced sweep operation outputs.

1. Configure the Sweep step using the **Sweep Configuration** page of the Step Setup tab for the step. To define a sweep operation output, you must specify at least one parameter in the **Sweepable parameters** field.
2. Switch to the **Sweep Output** page of the **Step Setup** tab for the Sweep step.
3. Click the **Add** button to display the Edit Sweep Output dialog box.
4. Place a checkmark in the **Advanced** checkbox. The **Output (Y-Axis)** and **Input X-Axis (Advanced)** fields display outputs of steps in the sweep operation.

Note The **Input X-Axis (Advanced)** field displays scalar outputs only. You cannot define a sweep operation output that compares two waveforms.

5. Select an output from the **Output (Y-Axis)** field. When you graph the sweep operation output, this value appears on the Y-axis of the graph.
6. Select an output from the **Input X-Axis (Advanced)** field. When you graph the sweep operation output, this value appears on the X-axis of the graph.
7. Click the **OK** button. The sweep operation output appears as an output of the Sweep step in the Project View.
8. Drag the sweep operation output to the Data View tab to plot the output on a graph. If the output is a group of data, you can plot an element of the group or the entire group.

Parallel and Nested Sweep Operations

You can use the Sweep step to perform a parallel sweep operation that sets values on multiple step parameters and multiple steps simultaneously. You also can perform a nested sweep operation by enclosing a Sweep step within another Sweep step.

Parallel Sweeps

Sweeping two or more step parameters simultaneously is called a parallel sweep. For example, if you want to vary the amplitude of a stimulus signal, you can maximize the precision of the acquisition by also varying the input range of the digitizer or MIO board that provides the signal. As the signal level gets higher, you can increase the input range of the measurement device to ensure you use the entire resolution for the measurement.

To configure a parallel sweep, navigate to the Step Setup tab for the Sweep step and add the parameters you want to sweep to the **Sweepable parameters** list on the **Sweep Configuration** page.

Nested Sweeps

You can perform nested sweeps to iterate through one range of values while you vary another range of values. For example, you might want to sweep through frequencies of a stimulus signal at different amplitudes. Using a nested sweep, you can set the amplitude to level 1 and sweep through the frequencies, then set the amplitude to level 2 and sweep through the frequencies, and so on.

To create a nested sweep, right-click a Sweep step in the Project View and select **Add nested sweep** from the shortcut menu. A new Sweep step appears and encloses the existing Sweep step. Configure the new Sweep step to sweep the parameter you want to change each time the original Sweep step completes a sweep operation.

Note You can control a step parameter with only one Sweep step at a time. If a parameter is swept by one Sweep step, the same or another Sweep step cannot sweep that parameter again in either a parallel or nested sweep.

Sweepable Parameters

The following table lists the parameters you can include in a sweep operation for each step.

Step Name	Parameter	Conditions
Create Analog Signal	Frequency	Only if **Signal type** is Sine Wave, Square Wave, Triangle Wave, Sawtooth Wave, or Formula
	Phase	Only if **Signal type** is Sine Wave, Square Wave, Triangle Wave, or Sawtooth Wave
	Amplitude	All signal types except DC Signal
	Offset	All signal types
	Duty cycle	Only if **Signal type** is Square Wave
	Sample rate	Always sweepable
	Block size	Always sweepable
	Start freq.	Only if **Signal type** is Multi-tone
	Stop freq.	Only if **Signal type** is Multi-tone
	Step freq.	Only if **Signal type** is Multi-tone
Create Digital Signal	Block size	Always sweepable
	Sample rate	Always sweepable
DAQmx Acquire	Samples to read	Always sweepable
	Sample rate	Always sweepable
	Input range max value (per channel)	Always sweepable
	Input range min value (per channel)	Always sweepable
NI-SCOPE Acquire	Sample Rate	Always sweepable
	Record Length	Always sweepable

	Channel **i** - Range	Only if channel **i** is enabled
	Channel **i** - Offset	Only if channel **i** is enabled
	Trigger Delay	Only if trigger **Type** is not Immediate
	Trigger Level	Only if trigger **Type** is Edge or Hysteresis
	Video Trigger Line Number	Only if trigger **Type** is Video
NI-FGEN Standard Function	Frequency	Always sweepable
	Duty Cycle	Only if function **Type** is Square
	Amplitude	Always sweepable
	Offset	Always sweepable
	Phase	Always sweepable
NI-FGEN Arbitrary Waveform	Sample Rate	Only if **Extract from waveform** is not selected
	Gain	Only if **Extract from waveform** is not selected
	Offset	Only if **Extract from waveform** is not selected
Filter	Cutoff	Only if **Type** is **Lowpass** or **Highpass**
	Low cutoff	Only when **Type** is **Bandpass** or **Bandstop**
	High cutoff	Only when **Type** is **Bandpass** or **Bandstop**
	Order	Only when **Mode** is **IIR Filter**
	Number of taps	Only when **Mode** is **FIR Filter**
Scaling and Conversion (Time Domain)	Pre-gain offset	Always sweepable

	Gain	Always sweepable
	Post-gain offset	Always sweepable
Scaling and Conversion (Frequency Domain–Magnitude)	Gain	Always sweepable
Scaling and Conversion (Frequency Domain–Phase)	Correction delay	Always sweepable
Subset and Resample	Start position	Only when you place a checkmark in the **Extract subset** checkbox
	Length	Only when you place a checkmark in the **Extract subset** checkbox
	dt	Only when you place a checkmark in the **Resample** checkbox
Interactive Alignment	x-offset	Only when **Mode** is **Manual** and you place a checkmark in the **Allow x-offset** checkbox
	y-offset	Only when **Mode** is **Manual** and you place a checkmark in the **Allow y-offset** checkbox
	x-gain	Only when **Mode** is **Manual** and you place a checkmark in the **Allow x-gain** checkbox
	y-gain	Only when **Mode** is **Manual** and you place a checkmark in the **Allow y-gain** checkbox
Power Spectrum	Number of avg.	Always sweepable
Frequency Response	Number of avg.	Always sweepable
Distortion	Highest harm.	Always sweepable

	Approx. fund. freq.	Always sweepable
Tone Extraction	Approx. freq.	Always sweepable
Limit Test	Upper constant	Only when **Limits source** is **User Defined Constants**
	Lower constant	Only when **Limits source** is **User Defined Constants**
	Upper gain	Only when **Limits window based on** is **Single Limit & Range**
	Upper offset	Only when **Limits window based on** is **Single Limit & Range**
	Lower gain	Only when **Limits window based on** is **Single Limit & Range**
	Lower offset	Only when **Limits window based on** is **Single Limit & Range**
	Limit constant	Only when **Limits window based on** is **Single Limit & Range**

The following steps do not contain parameters you can include in a sweep operation:

- Load from ASCII
- Load from LVM
- Load from SPICE
- Save to ASCII/LVM
- Time Averaging
- Window
- Arithmetic
- Formula (Scalar)
- Amplitude and Levels
- Histogram
- Timing and Transition

Note Sweeping NI-FGEN Standard Function parameters does not cause the Start Trigger to be resent for each iteration. The device may reset on each iteration, depending on which parameters you are sweeping. During this device reset, the RTSI lines may be reset, resulting in what looks like a false trigger. This false trigger does not correlate with the timing of the start of the generation. If you are triggering an NI-SCOPE Acquire or DAQmx Acquire step, consider making the NI-SCOPE Acquire or DAQmx Acquire step execute after the NI-FGEN Standard Function step by placing a checkmark in the **Start this step after** checkbox on the **Execution Control** tab, and selecting the NI-FGEN Standard Function step as the step on which to wait.

Sequencing Steps to Reuse Hardware

When you configure a step to use a hardware device, the step reserves the device so that no other step or other application can use the device. However, you might want to reuse the same device in one project. You can use a Sequence step to pause the execution of other steps in the project while one step uses the device, thereby allowing multiple steps in the same project to use the same hardware.

For example, if you are using a data acquisition device and you want to acquire signals from different channels at different rates, you can use the Sequence step to acquire a signal from one channel at one rate and then pause that acquisition while you acquire a signal from another channel at another rate. If you try to set up the same project without a Sequence step, LabVIEW SignalExpress returns an error because only one executing step can use a single hardware device at a time.

Note When you use the Sequence step to reuse hardware, you cannot perform a continuous signal acquisition because LabVIEW SignalExpress stops and starts the hardware device.

Complete the following steps to allow multiple steps to use the same hardware.

1. Install the hardware device, connect it to the PC, and configure LabVIEW SignalExpress to begin acquiring signals from the hardware device.
2. Add a Sequence step to the Project View after the Acquire Signals step.
3. On the Step Setup tab for the Sequence step, select **Run preceding steps before following steps** and place a checkmark in the **Allow hardware reuse** checkbox.
4. Add another Acquire Signals step to the Project View.
5. Configure the second Acquire Signals step to acquire a signal from the same hardware device as the first Acquire Signals step.
6. Click the **Run** button to run the project. For each iteration of the project, LabVIEW SignalExpress runs the first Acquire Signals step once, pauses the execution of that step and reconfigures the hardware device for the second acquisition, runs the second Acquire Signals step once, then pauses the execution of that step

and reconfigures the hardware device for the first step.

Data Logging

You can log time waveform, scalar, or Boolean signals you create or acquire in LabVIEW SignalExpress, and you can use steps to analyze and process logged signals. Use data logging to save, review, and analyze measurements you take in LabVIEW SignalExpress.

Note LabVIEW SignalExpress also supports limited logging of frequency-domain signals.

Logging a Signal

If a project includes steps with valid output signals, you can begin logging immediately by clicking the **Record** button and selecting a signal from the **Logging Signals Selection** dialog box. LabVIEW SignalExpress runs the project and logs the signal continuously until you click the **Stop** button, and the log appears in the Logged Data window at the bottom of the Project View. LabVIEW SignalExpress logs signals in blocks that are equivalent to the block size, or number of samples, of the acquired signal. You configure the block size of a signal in the step you use to create or acquire the signal.

Like a step output, you can drag a logged signal from the **Logged Data** window to the Data View tab to display the data. You can scroll through the logged signal and view any alarms or events associated with the log. You also can switch to the Playback work area to play back and analyze logged signals.

Note You must be in the Monitor/Record work area to log a signal.

LabVIEW SignalExpress saves logged data in the .tdms file format to the directory you specify on the Logging page of the Options dialog box. The .tdms file is a binary file that contains waveform data and stores waveform properties. You also can import logged data from .tdms files you previously created with LabVIEW SignalExpress.

Logging While Running

If you want to log a signal every time you run a project, you can specify the signal(s) to record in one of the following ways:

- In the **Project View**, right-click a step output and select **Enable Recording** from the shortcut menu to enable data logging for the output.
- Select **View»Recording Options** to display the Recording Options tab and select a signal or signals on the Signal Selection page.

Note The **Recording Options** tab also includes pages you can use to configure advanced logging options, such as start and stop conditions, alarms, and events.

When you use one of the previous methods to select a signal or signals to record, the **Record While Running** button appears in the toolbar. Ensure that the **Record While Running** button is enabled, and click the **Run** button to begin logging the signals you selected.

Logging Frequency-Domain Signals

You can log the last known value of a frequency-domain signal and view the log on the **Data View** tab. The last known value of the signal is the value of the signal when the project stops running. In the Project View, right-click a frequency-domain output signal and select **Record last value** from the shortcut menu to log the last known value of the signal. LabVIEW SignalExpress saves logs of frequency-domain signals in the .tdms file format to the default location you specify on the Logging page of the Options dialog box.

Note Because LabVIEW SignalExpress records a frequency-domain signal as a single value, you cannot play back a log of a frequency-domain signal.

Managing Multiple Data Logs

When you record multiple logs, the most recent log becomes the active log. If you are viewing the signal(s) you are logging on the **Data View** tab, the display updates to display the new active log. The name of the active log also appears in bold in the **Logged Data** window. If you are in the Playback work area, any steps that are using the logged signal as an input update to use the new active log as the input. You can change the active log at any time by right-clicking the name of a log in the **Logged Data** window and selecting **Make Active Log** from the shortcut menu.

Logging a Signal

Complete the following steps to create a log of a specified signal.

1. Place the Create Analog Signal step in the Project View.
2. In the **Signal type** pull-down menu, select **Noise Signal**.
3. If the **Recording Options** tab is not visible, select **View»Recording Options** to display the tab and the data logging configuration options.
4. Select the **Signal Selection** option in the **Category** list to display the Signal selection page. The **Signal selection** page displays the signals that are available to log. Place a checkmark in the checkbox next to the white Gaussian noise signal.

   When you select to record the signal, the **Record** button on the toolbar changes to the **Record While Running** button. When the **Record While Running** button is pressed, you can press the **Run** button to log the selected signal.
5. Select the **Log Summary** option in the **Category** list to display the Log Summary page. Specify the name of the log in the **Log title** text box and provide a description of the log in the **Log description** text box.
6. Click the **Run** button on the toolbar to begin logging the signal.

Note Changing the configuration of a LabVIEW SignalExpress project during a data logging operation stops the current data logging operation.

7. When you are ready to stop logging the signal, click the **Stop** button on the toolbar.

Note You also can configure logging to start and stop according to user-defined conditions. Use the Start Conditions page to configure logging start conditions and the Stop Conditions page to configure logging stop conditions.

The logged signal appears in the Logged Data window. To view a logged signal, drag the logged signal to the Data View. You also can analyze the logged signal using analysis steps by switching to the Playback work area to process a specific log.

Alarms and Events

You can use the Alarms and Events pages of the **Recording Options** tab to specify conditions under which LabVIEW SignalExpress records an alarm or an event in a logged signal.

Alarms

An alarm is a notification of a signal state. You can configure alarms to activate when a signal is greater than a specified value, less than a specified value, or within or outside of a specified range of values. For Boolean signals, you can configure an alarm to activate when the signal becomes TRUE or FALSE.

LabVIEW SignalExpress automatically records changes in the alarm state, or when an alarm activates or deactivates, in the log file. When you drag a logged signal to the Data View tab, the graph displays markers at the locations where the alarm state changes during the logging operation. You also can configure actions for LabVIEW SignalExpress to take when an alarm state changes. For example, you can generate a sound, display a message to the user, generate a software trigger, create a snapshot, or execute a command to run a program.

Events

An event is an occurrence at a specific point in time. You can configure LabVIEW SignalExpress to acknowledge keystroke or signal-based events. For keystroke events, LabVIEW SignalExpress records an event in the log when a user presses the key or combination of keys you specify. For signal-based events, LabVIEW SignalExpress records an event in the log when a source signal shows a rising slope, a falling slope, or when the signal enters or leaves a window of values you specify. Use events when you want to note an occurrence during logging. For example, if you are logging sound pressure in an area with outside noise, you can configure a keystroke event so that a user can press the key each time a noise occurs that is not relevant to the signal you are recording.

LabVIEW SignalExpress automatically records events in the log file, and you can specify an annotation to display on the graph on the **Data View** tab when the event occurs. You also can prompt the user to enter an annotation at the time a keystroke event occurs.

Viewing Alarms and Events

LabVIEW SignalExpress displays alarm state changes and events on the graph of a logged signal, but alarm state changes and events might be difficult to locate in long logs. You can use the Event Viewer to navigate to and zoom in on alarm state changes and events. Right-click a graph on the **Data View** tab and select **Visible Items»Event Viewer** from the shortcut menu to display the Event Viewer. The Event Viewer displays similar information to the Event Log tab, and lists each alarm state change and event that occurred, as well as errors, warning, data loss notifications, and informational messages. Double-click an alarm state change or event in the Event Viewer to zoom in on the portion of the graph where the alarm state change or event occurred. You also can use the Preview graph to zoom in on a portion of a logged signal and view a specific alarm state change or event.

If you want to see a list of alarm state changes and events that you can export to a text file, right-click a logged signal in the Logged Data window and select **Show Alarms and Events** from the shortcut menu. Click the **Export** button in the window that appears to save the list as a text file.

Importing Logged Data

You can import logged signals into a LabVIEW SignalExpress project from another LabVIEW SignalExpress project or from a .tdms file you previously created in LabVIEW SignalExpress. Select **File»Import»Logged Signals From Another Project** to select a project file from which to import logged data. LabVIEW SignalExpress adds all the logged signals from the project you select to the Logged Data window of the current project.

Select **File»Import»Logged Signals from SignalExpress TDMS Files** to select a .tdms file from which to import logged data. LabVIEW SignalExpress adds the log associated with the .tdms file to the **Logged Data** window of the current project.

Note You can select an individual .tdms file or a folder that contains the .tdms file to import the logged data. LabVIEW SignalExpress imports all the information associated with the logged data regardless of whether you select the file or a folder that contains the file.

Converting Logs to ASCII Files

You can convert logged signals in LabVIEW SignalExpress to an ASCII format text file. The ASCII file includes the date, start time, and name of the log and lists the values of every data point logged for every signal in the log. To convert a log to an ASCII file, right-click a signal or group of signals in the Logged Data window and select **Convert to ASCII** from the shortcut menu. LabVIEW SignalExpress prompts you for a name and location to use to save the text file.

Note LabVIEW SignalExpress supports importing ASCII files as live data only. You cannot import data in an ASCII file back into LabVIEW SignalExpress as a log. If you want to import logged data into LabVIEW SignalExpress, you must import the data from another LabVIEW SignalExpress project or from a .tdms file.

Exporting Logs to Microsoft Excel

You can export logged signals from LabVIEW SignalExpress to Microsoft Excel. LabVIEW SignalExpress exports the date, start time, and name of the log, and lists the values of every data point logged for every signal in the log. To export a log to Microsoft Excel, right-click a signal or group of signals in the Logged Data window and select **Export to Microsoft Excel** from the shortcut menu.

Note When you export data to Microsoft Excel, LabVIEW SignalExpress copies data samples to a clipboard for export. Use the **Maximum Clipboard Data Export Size** option on the Data page of the Options dialog box to increase the number of data samples the clipboard can contain. Depending on the amount of memory on a machine, large log files might not export successfully.

If Microsoft Excel is not running when you export a log file, LabVIEW SignalExpress opens Microsoft Excel and displays the exported data in a new book file. If Microsoft Excel is running when you export a log file, LabVIEW SignalExpress adds worksheets containing the exported data to the active book file. If a log contains multiple signals, each signal appears on a separate worksheet.

Note LabVIEW SignalExpress supports importing logged data from .tdms files or other LabVIEW SignalExpress projects only. You cannot import logged data from Microsoft Excel.

LabVIEW SignalExpress does not save the Microsoft Excel file. You must save the exported file manually in Microsoft Excel.

Running Projects in Operator Mode

Operator mode provides a way to disable editing capabilities for LabVIEW SignalExpress projects. Disabling editing capabilities can be useful if you want to distribute a project to other users. For example, if you create a project that generates a signal, you can configure the project so that in operator mode a user only can adjust the frequency of the signal. You also can use operator mode to create an interface with a limited set of controls that is easy to use and learn.

You can set a password on a project so that a user must enter the password to disable operator mode. If you save a project in operator mode, the project always opens in operator mode unless you save the project again with operator mode disabled.

If you want to allow a user to change signal values on a project in operator mode, you must create an operator interface. An operator interface contains controls that you bind to specific step parameters so that in operator mode, changing the value of the control changes the value of the step parameter. Use the Operator Interface view and the Toolbox and Properties windows to create an operator interface.

Note If you save a project in operator mode, LabVIEW SignalExpress saves the project with its initial signal values, regardless of whether you change the value with a control.

After you configure a project with an operator interface, select **Operate»Operator Mode»Operator Mode Enabled** or click the **Operator Mode Enabled** button on the Operator Interface view to enable operator mode. When a project is in operator mode, only the **Run** and **Record** buttons appear in the toolbar, and a limited set of menus and menu options are available. You cannot display any additional tabs or views when a project is in operator mode, and you cannot add, delete, or configure steps. Configure LabVIEW SignalExpress to display all the tabs, views, and steps you need before you enable operator mode.

If a project is in operator mode, selecting **Operate»Operator Mode»Operator Mode Enabled** or clicking the **Operator Mode Enabled** button on the Operator Interface view disables operator mode.

Setting an Operator Mode Password

Before you enable operator mode, you can specify a password that a user must enter to disable operator mode. Select **Operate»Operator Mode»Set Operator Mode Password** or click the **Set Operator Mode Password** button on the Operator Interface view to set an operator mode password. When you save the project, LabVIEW SignalExpress saves the password with the project.

Note LabVIEW SignalExpress does not encrypt operator mode passwords. Use operator mode passwords for simple access control.

Creating an Operator Interface

When a project is in operator mode, LabVIEW SignalExpress allows limited interaction with the project. For example, you cannot change configuration options on the Step Setup tab when a project is in operator mode. However, if you distribute a project saved in operator mode to other users, you might want to allow the users to modify certain step configuration settings. You can allow limited editing by configuring an operator interface with controls bound to various step parameters before you enable operator mode.

Note You cannot configure an operator interface when a project is in operator mode.

Complete the following steps to configure an operator interface.

1. If the Operator Interface view is not visible, select **View»Operator Interface** to display the Operator Interface view.
2. If the Toolbox window is not visible, select **View»Toolbox** or right-click in the Operator Interface view and select **Toolbox** from the shortcut menu to display the **Toolbox** window. The **Toolbox** window displays a list of controls you can add to the Operator Interface view.
3. Select a control in the **Toolbox** window and drag the control to the Operator Interface view. The control appears with resizing handles and a small arrow icon ().
4. Click the small arrow icon to display a **Tasks** window you can use to configure basic properties of the control, such as a caption, initial value, and range of selectable values.
5. In the **Tasks** window, click the **Edit Bound Parameters** link to bind the control to parameters of steps in the project.
6. (Optional) Select **View»Properties** or right-click a control and select **Properties** from the shortcut menu to display the Properties window. Use the **Properties** window and configure additional properties of a control. For example, the **Properties** window displays additional appearance and behavior configuration properties.
7. Click the **Operator Mode Enabled** button on the Operator Interface view or select **Operate»Operator Mode»Operator**

Mode Enabled to enable operator mode. Notice that a limited set of toolbar buttons and menu items appear in operator mode.

8. In the Project View, find a step that contains a bound parameter, drag the output of the step to the **Data View** tab, and run the project.
9. While the project is running, use the control to see how changing the value of the bound parameter changes the value of the signal.
10. Save the project.

Binding Operator Interface Controls to Step Parameters

When you enable operator mode, LabVIEW SignalExpress limits the amount of interaction users can have with a project. However, you can create an operator interface that allows a user to adjust the values of certain step parameters when operator mode is enabled. You create an operator interface by adding controls to the Operator Interface view, and you enable those controls to adjust step parameter values by binding the controls to a specific parameter or set of parameters.

Note You cannot bind text or label controls to step parameters. You can bind a single ring control to one parameter only.

Complete the following steps to bind an operator interface control to a step parameter.

1. Select a control on the operator interface. Resizing handles and a small arrow icon () appear.
2. Click the small arrow icon to display the **Tasks** window.
3. In the **Tasks** window, click the **Edit Bound Parameters** link to display the Bound Parameters List dialog box. This dialog box binds the control to parameters of steps in the project.

Note You also can display the **Bound Parameters List** dialog box from the Properties window. Display the **Properties** window, select a control on the operator interface, and select the Bound Parameters property that appears under **Parameter Binding** in the **Properties** window. Click the button that appears next to the value of Bound Parameters to display the **Bound Parameters List** dialog box.

4. In the **Bound Parameters List** dialog box, click the **Add** button under the **Members** list to display the **Bound Parameter Editor** window, which displays a list of all the steps in the project.
5. In the **Bound Parameter Editor** window, click the expand icon next to a step to display the parameters of the step to which you can bind the control.
6. Select a parameter and click the **OK** button. The parameter appears in the **Members** list of the **Bound Parameters List**

dialog box. The **Properties** list also updates to display properties of the parameter you select. Expand a property in the **Properties** list to edit the property. You also can use the **Scale** option to specify how changing the value of the control changes the value of the property. For example, if you specify a **Multiplier** of 2, incrementing the control value by 1 actually increments the property value by 2.

7. (Optional) Repeat steps 4 through 6 to bind the control to additional parameters.
8. Click the **OK** button to close the **Bound Parameters List** dialog box and bind the control to the parameter(s).

Background Color

Specifies the background color of the operator interface.

Foreground Color

Specifies the foreground color of the operator interface.

Name

Specifies the name of the operator interface. The name appears on the title bar of the operator interface.

Knob Properties

You can use the following properties to configure Knob controls on an operator interface.

Property	Description
Background Color	Specifies the color to use for the background of the control. Details
Border	Specifies the type of border to use for the control. Details
Bound Parameters	Specifies the step parameters that are bound to the control. Click the button that appears when you select the value of this property to display the Bound Parameters List dialog box. Details
Coercion Interval	Specifies the interval to use to coerce the value of the control. Details
Coercion Mode	Specifies the mode to use to coerce the value of the control. Details
Dial Color	Specifies the color to use for the dial of a knob control. Details
Foreground Color	Specifies the color to use for the foreground of the control. Details
Initial Value	Specifies the initial value of the control. Details
Interaction Mode	Specifies the types of interactions the user can have with the control when you run the project in operator mode. Details
Knob Style	Specifies the style of the knob control. Details
Pointer Color	Specifies the color to use for the pointer of the control. Details
Range	Specifies the minimum and maximum values of the control. Details
Range: Maximum	Specifies the maximum value of the control. Details
Range: Minimum	Specifies the minimum value of the control. Details
Scale Arc	Specifies the arc on which the scale is drawn. Details

Scale Base Line Color	Specifies the color of the line that connects the tick marks on the scale. You must set the Scale Base Line Visible property to **True** for the base line to appear. Details
Scale Base Line Visible	Specifies whether the line that connects the tick marks on the scale is visible. Details
Scale Type	Specifies the type of the scale. Details
Scale Visible	Specifies whether the scale of the control is visible. Details

Background Color

[Control types: Knob, Numeric, Slide, Switch, Ring, Text, Label] Specifies the color to use for the background of the control.

Border

[Control types: Knob, Slide, Switch] Specifies the type of border to use for the control.

Bound Parameters

[Control types: Knob, Numeric, Slide, Switch] Specifies the step parameters that are bound to the control. Click the button that appears when you select the value of this property to display the Bound Parameters List dialog box.

Coercion Interval

[Control types: Knob, Numeric, Slide] Specifies the interval to use to coerce the value of the control.

Coercion Mode

[Control types: Knob, Numeric, Slide] Specifies the mode to use to coerce the value of the control.

Dial Color

[Control types: Knob] Specifies the color to use for the dial of a knob control.

Foreground Color

[Control types: Knob, Numeric, Slide, Switch, Ring, Text, Label] Specifies the color to use for the foreground of the control.

Initial Value

[Control types: Knob, Numeric, Slide] Specifies the initial value of the control.

Interaction Mode

[Control types: Knob, Numeric, Slide, Switch] Specifies the types of interactions the user can have with the control when you run the project in operator mode.

Knob Style

[Control types: Knob] Specifies the style of the knob control.

Pointer Color

[Control types: Knob, Slide] Specifies the color to use for the pointer of the control.

Range

[Control types: Knob, Numeric, Slide] Specifies the minimum and maximum values of the control.

Range: Maximum

[Control types: Knob, Numeric, Slide] Specifies the maximum value of the control.

Range: Minimum

[Control types: Knob, Numeric, Slide] Specifies the minimum value of the control.

Scale Arc

[Control types: Knob] Specifies the arc on which the scale is drawn.

Scale Base Line Color

[Control types: Knob, Slide] Specifies the color of the line that connects the tick marks on the scale. You must set the Scale Base Line Visible property to **True** for the base line to appear.

Scale Base Line Visible

[Control types: Knob, Slide] Specifies whether the line that connects the tick marks on the scale is visible.

Scale Type

[Control types: Knob, Slide] Specifies the type of the scale.

Scale Visible

[Control types: Knob, Slide] Specifies whether the scale of the control is visible.

Label Properties

You can use the following properties to configure Label controls on an operator interface.

Property	Description
Background Color	Specifies the color to use for the background of the control. Details
Border Style	Specifies the style of the border of the control. Details
Font	Specifies the font to use for the text in the control. Details
Font: Bold	Specifies whether the text in the control appears bolded. Details
Font: GdiCharSet	Specifies the GDI (Graphics Device Interface) character set that the font of the text in the control uses. Details
Font: GdiVerticalFont	Specifies whether the font of the text in the control is derived from a GDI (Graphics Device Interface) vertical font. Details
Font: Italic	Specifies whether the text in the control appears italicized. Details
Font: Name	Specifies the name of the font to use for the text in the control. Details
Font: Size	Specifies the size of the font to use in the control. Details
Font: Strikeout	Specifies whether a horizontal line appears through the text in the control. Details
Font: Underline	Specifies whether the text in the control appears underlined. Details
Font: Unit	Specifies the unit of measure for the size of the font you use in the control. Details
Foreground Color	Specifies the color to use for the foreground of the control. Details
Text	Specifies the text that appears in the control. Details

Border Style

[Control types: Numeric, Ring, Text, Label] Specifies the style of the border of the control.

Font

[Control types: Text, Label] Specifies the font to use for the text in the control.

Font: Bold

[Control types: Text, Label] Specifies whether the text in the control appears bolded.

Font: GdiCharSet

[Control types: Text, Label] Specifies the GDI (Graphics Device Interface) character set that the font of the text in the control uses.

Font: GdiVerticalFont

[Control types: Text, Label] Specifies whether the font of the text in the control is derived from a GDI (Graphics Device Interface) vertical font.

Font: Italic

[Control types: Text, Label] Specifies whether the text in the control appears italicized.

Font: Name

[Control types: Text, Label] Specifies the name of the font to use for the text in the control.

Font: Size

[Control types: Text, Label] Specifies the size of the font to use in the control.

Font: Strikeout

[Control types: Text, Label] Specifies whether a horizontal line appears through the text in the control.

Font: Underline

[Control types: Text, Label] Specifies whether the text in the control appears underlined.

Font: Unit

[Control types: Text, Label] Specifies the unit of measure for the size of the font you use in the control.

Text

[Control types: Ring, Text, Label] Specifies the text that appears in the control.

Numeric Properties

You can use the following properties to configure Numeric controls on an operator interface.

Property	Description
Background Color	Specifies the color to use for the background of the control. Details
Border Style	Specifies the style of the border of the control. Details
Bound Parameters	Specifies the step parameters that are bound to the control. Click the button that appears when you select the value of this property to display the Bound Parameters List dialog box. Details
Coercion Interval	Specifies the interval to use to coerce the value of the control. Details
Coercion Mode	Specifies the mode to use to coerce the value of the control. Details
Foreground Color	Specifies the color to use for the foreground of the control. Details
Format Mode	Specifies the numeric format to use for the value of the control. Details
Initial Value	Specifies the initial value of the control. Details
Interaction Mode	Specifies the types of interactions the user can have with the control when you run the project in operator mode. Details
Range	Specifies the minimum and maximum values of the control. Details
Range: Maximum	Specifies the maximum value of the control. Details
Range: Minimum	Specifies the minimum value of the control. Details
Text Alignment	Specifies the alignment of text within a numeric control. Details
Up/Down	Specifies the alignment of the up and down arrows in the

Alignment	control relative to the text box. Details

Format Mode

[Control types: Numeric] Specifies the numeric format to use for the value of the control.

Text Alignment

[Control types: Numeric] Specifies the alignment of text within a numeric control.

Up/Down Alignment

[Control types: Numeric] Specifies the alignment of the up and down arrows in the control relative to the text box.

Ring Properties

You can use the following properties to configure Ring controls on an operator interface.

Property	Description
Background Color	Specifies the color to use for the background of the control. Details
Bound Parameter	Specifies the step parameters that are bound to the control. Click the button that appears when you select the value of this property to display the Bound Parameters List dialog box. Details
Foreground Color	Specifies the color to use for the foreground of the control. Details

Bound Parameter

[Control types: Ring] Specifies the step parameter that is bound to the control. Click the button that appears when you select the value of this property to display the Bound Parameters List dialog box.

Slide Properties

You can use the following properties to configure Slide controls on an operator interface.

Property	Description
Background Color	Specifies the color to use for the background of the control. Details
Border	Specifies the type of border to use for the control. Details
Bound Parameters	Specifies the step parameters that are bound to the control. Click the button that appears when you select the value of this property to display the Bound Parameters List dialog box. Details
Coercion Interval	Specifies the interval to use to coerce the value of the control. Details
Coercion Mode	Specifies the mode to use to coerce the value of the control. Details
Fill Background Color	Specifies the color of the unfilled portion of the fill area of a slide control. Details
Fill Base Value	Specifies the value at which the fill begins. You must specify a base value if you set the Fill Mode property to **ToBaseValue**. Details
Fill Color	Specifies the color of the filled portion of the fill area of a slide control. Details
Foreground Color	Specifies the color to use for the foreground of the control. Details
Initial Value	Specifies the initial value of the control. Details
Interaction Mode	Specifies the types of interactions the user can have with the control when you run the project in operator mode. Details
Inverted Scale	Specifies whether to invert the scale of the control. Details
Range	Specifies the minimum and maximum values of the control. Details

Range: Maximum	Specifies the maximum value of the control. Details
Range: Minimum	Specifies the minimum value of the control. Details
Scale Base Line Color	Specifies the color of the line that connects the tick marks on the scale. You must set the Scale Base Line Visible property to **True** for the base line to appear. Details
Scale Base Line Visible	Specifies whether the line that connects the tick marks on the scale is visible. Details
Scale Type	Specifies the type of the scale. Details
Scale Visible	Specifies whether the scale of the control is visible. Details
Slide Style	Specifies the style of the slide control. Details

Fill Background Color

[Control types: Slide] Specifies the color of the unfilled portion of the fill area of a slide control.

Fill Base Value

[Control types: Slide] Specifies the value at which the fill begins. You must specify a base value if you set the Fill Mode property to **ToBaseValue**.

Fill Color

[Control types: Slide] Specifies the color of the filled portion of the fill area of a slide control.

Fill Mode

[Control types: Slide] Specifies the direction in which the control fills. Controls fill from the minimum value to the slider value by default.

Fill Style

[Control types: Slide] Specifies the style to use to draw the fill color.

Inverted Scale

[Control types: Slide] Specifies whether to invert the scale of the control.

Slide Style

[Control types: Slide] Specifies the style of the slide control.

Switch Properties

You can use the following properties to configure Switch controls on an operator interface.

Property	Description
Background Color	Specifies the color to use for the background of the control. Details
Bound Parameters	Specifies the step parameters that are bound to the control. Click the button that appears when you select the value of this property to display the Bound Parameters List dialog box. Details
Foreground Color	Specifies the color to use for the foreground of the control. Details
Initial Value	Specifies the initial value of the control. Details
Interaction Mode	Specifies the types of interactions the user can have with the control when you run the project in operator mode. Details
Off Color	Specifies the color of a switch control when the switch is in the off or FALSE position. Details
On Color	Specifies the color of a switch control when the switch is in the on or TRUE position. Details
Switch Style	Specifies the style of a switch control. Details

Off Color

[Control types: Switch] Specifies the color of a switch control when the switch is in the off or FALSE position.

On Color

[Control types: Switch] Specifies the color of a switch control when the switch is in the on or TRUE position.

Switch Style

[Control types: Switch] Specifies the style of a switch control.

Text Properties

You can use the following properties to configure Text controls on an operator interface.

Property	Description
Background Color	Specifies the color to use for the background of the control. Details
Border Style	Specifies the style of the border of the control. Details
Font	Specifies the font to use for the text in the control. Details
Font: Bold	Specifies whether the text in the control appears bolded. Details
Font: GdiCharSet	Specifies the GDI (Graphics Device Interface) character set that the font of the text in the control uses. Details
Font: GdiVerticalFont	Specifies whether the font of the text in the control is derived from a GDI (Graphics Device Interface) vertical font. Details
Font: Italic	Specifies whether the text in the control appears italicized. Details
Font: Name	Specifies the name of the font to use for the text in the control. Details
Font: Size	Specifies the size of the font to use in the control. Details
Font: Strikeout	Specifies whether a horizontal line appears through the text in the control. Details
Font: Underline	Specifies whether the text in the control appears underlined. Details
Font: Unit	Specifies the unit of measure for the size of the font you use in the control. Details
Foreground Color	Specifies the color to use for the foreground of the control. Details
Read Only	Specifies whether you can edit the text that appears in the control. Details
Text	Specifies the text that appears in the control. Details

Read Only

[Control types: Text] Specifies whether you can edit the text that appears in the control.

Performing Common Tasks in LabVIEW SignalExpress

This book contains example procedures that guide you through some common tasks in LabVIEW SignalExpress. The examples in this book assume you have NI-DAQmx installed, and that you have configured an installed or simulated NI-DAQmx device.

Note You can install NI-DAQmx software from the National Instruments Device Driver DVD, or you can download the latest version of NI-DAQmx software from ni.com/downloads.

This book includes the following examples:

- Acquiring, Logging, and Analyzing a Temperature Signal

Acquiring, Logging, and Analyzing a Temperature Signal

The examples in this book describe how to acquire, log, and analyze a temperature signal from an NI-DAQmx device. The examples build on each other, so you must start with the first example in order to build the project correctly. The examples in this book assume you have NI-DAQmx installed, and that you have configured an installed or simulated NI-DAQmx device.

Note You can install NI-DAQmx software from the National Instruments Device Driver DVD, or you can download the latest version of NI-DAQmx software from ni.com/downloads.

While the examples in this book are specific to acquiring a temperature signal, you can apply the concepts from these procedures to any signal you acquire, log, and/or analyze in LabVIEW SignalExpress. This book contains the following example procedures:

- Example 1: Logging a Temperature Signal
- Example 2: Logging a Temperature Signal with Start and Stop Conditions
- Example 3: Displaying Alarms when a Temperature Signal Meets a Specified Value
- Example 4: Analyzing a Logged Temperature Signal

Example 1: Logging a Temperature Signal

You can use LabVIEW SignalExpress to log signals you acquire from various devices and instruments. The following procedure describes how to log a temperature signal from an NI-DAQmx device. This example assumes you have a thermocouple temperature sensor connected to physical channel ai0 of an NI-DAQmx device.

Note You can install NI-DAQmx software from the National Instruments Device Driver DVD, or you can download the latest version of NI-DAQmx software from ni.com/downloads.

Complete the following steps to use LabVIEW SignalExpress to acquire a temperature signal from the device:

1. Launch LabVIEW SignalExpress and select **File»New Project** to open a new LabVIEW SignalExpress project.
2. Select **Add Step»Acquire Signals»DAQmx Acquire»Analog Input»Temperature»Thermocouple** to add the DAQmx Acquire step to the Project View. The **Add Channels To Task** dialog box appears.
3. In the **Add Channels To Task** dialog box, select **ai0** under **Dev1 (*Device Name*)** and click the **OK** button. The Step Setup tab updates to display **Dev1_ai0** in the list of channels, and **Thermocouple Setup** configuration options appear to the right of the list of channels.
4. Use the **Thermocouple Setup** options to configure the step. Select the **Thermocouple Type** you are using, and select **deg F** from the **Scaled Units** pull-down menu to specify to measure the temperature in degrees Fahrenheit.
5. After you configure the step, click the **Record** toolbar button. The **Logging Signals Selection** dialog box appears.
6. In the **Logging Signals Selection** dialog box, expand **Thermocouple** in the **Signals to include** tree and place a checkmark in the **Dev1_ai0** checkbox.
7. Click the **OK** button to close the dialog box and begin recording the signal. A new log appears in the Logged Data window.
8. Switch to the Data View tab and drag the log from the **Logged Data** window to the **Data View** tab to view the progress of the

log.

Note You also can drag the **Thermocouple** output of the DAQmx Acquire step to the **Data View** tab to display the current value of the signal.

9. Click the **Record** or the **Stop** button to stop logging the signal.
10. Select **File»Save Project** to save the project.

Note When you save a project that contains logs, LabVIEW SignalExpress saves the logs with the project. LabVIEW SignalExpress also saves log files in the .tdms file format to a location you specify on the Logging page of the Options dialog box.

This example describes how to start and stop logging by clicking a button. However, you might want to start and stop logging based on when a signal meets certain conditions. Example 2: Logging a Temperature Signal with Start and Stop Conditions describes how to configure logging with start and stop conditions in LabVIEW SignalExpress.

Example 2: Logging a Temperature Signal with Start and Stop Conditions

The following procedure describes how to log a temperature signal from an NI-DAQmx device. This example uses the project you created in Example 1: Logging a Temperature Signal.

Complete the following steps to configure LabVIEW SignalExpress to begin logging the temperature signal when the signal rises above 75 degrees Fahrenheit and to stop logging the signal when the signal falls back below 75 degrees Fahrenheit.

1. Open the project you saved in *Example 1: Logging a Temperature Signal*.
2. Select **View»Recording Options** to display the Recording Options tab. The **Category** list displays the pages of the tab you can use to configure logging operations. The Signal Selection page is selected by default because you must select a signal to log before you can configure logging.
3. The **Signal selection** list on the **Signal Selection** page displays the signals in the project that you can log. In the **Channel Name** column, expand **Thermocouple** and place a checkmark in the **Record** checkbox for channel **Dev1/ai0**. The **Record While Running** button replaces the **Record** button in the toolbar.
4. In the **Category** list, select **Start Conditions** to display the Start Conditions page.
5. Click the **Add** button to add a new start condition to the **Start condition list** and display additional configuration options.
6. Verify that **Signal trigger** appears in the **Condition type** pull-down menu, **Thermocouple - Dev1_ai0** appears in the **Signal** pull-down menu, and **Rising slope** appears in the **Trigger type** pull-down menu.
7. Enter 75 in the **Trigger value** field to specify for logging to start when the signal from the thermocouple crosses 75 with a rising slope, or exceeds 75 degrees Fahrenheit.
8. In the **Category** list, select **Stop Conditions** to display the Stop Conditions page.
9. Click the **Add** button to add a new stop condition to the **Stop**

condition list and display additional configuration options.

10. Select **Signal trigger** from the **Condition type** pull-down menu.
11. Verify that **Thermocouple - Dev1_ai0** appears in the **Signal** pull-down menu and select **Falling slope** from the **Trigger type** pull-down menu.
12. Enter 75 in the **Trigger value** field to specify for logging to stop when the signal from the thermocouple crosses 75 with a falling slope, or falls below 75 degrees Fahrenheit.
13. Switch to the Data View tab and click the **Run** button to run the project continuously. The status indicator below the Project View displays **Waiting for start conditions** until LabVIEW SignalExpress detects that the temperature signal is above 75 degrees Fahrenheit.
14. After the start condition is met, the status indicator updates to display **Recording**. If the logged signal does not appear on the **Data View** tab, drag the signal from the Logged Data window to the **Data View** tab to view the progress of the log.
15. The project runs and LabVIEW SignalExpress logs the signal until the temperature falls below 75 degrees Fahrenheit and the stop condition is met. After the stop condition is met, LabVIEW SignalExpress stops recording the signal and stops running the project. Select **File»Save Project** to save the project.

In addition to start and stop conditions, you can use the **Recording Options** tab to configure LabVIEW SignalExpress to record alarms and events that occur during logging. Example 3: Displaying Alarms when a Temperature Signal Meets a Specified Value describes how to configure alarms in LabVIEW SignalExpress.

Example 3: Displaying Alarms when a Temperature Signal Meets a Specified Value

The following procedure describes how to display an alarm when a temperature signal you are logging exceeds a temperature you specify. This example uses the project you created in Example 1: Logging a Temperature Signal and modified in Example 2: Logging a Temperature Signal with Start and Stop Conditions.

Complete the following steps to configure LabVIEW SignalExpress to display an alarm when the temperature signal rises above 85 degrees Fahrenheit.

1. Open the project you saved in *Example 2: Logging a Temperature Signal with Start and Stop Conditions*.
2. Select **View»Recording Options** to display the Recording Options tab.
3. In the **Category** list, select **Alarms** to display the Alarms page.
4. Click the **Add** button to add a new alarm to the **Alarm list** and display additional configuration options.
5. Verify that **Thermocouple - Dev1_ai0** appears in the **Signal** pull-down menu and **Above** appears in the **Condition** pull-down menu.
6. Enter 85 in the **Value** field to specify that LabVIEW SignalExpress displays an alarm when the signal from the thermocouple rises above 85 degrees Fahrenheit.
7. Switch to the Data View tab and click the **Run** button to run the project continuously. The status indicator below the Project View displays **Waiting for start conditions** until LabVIEW SignalExpress detects the start condition you configured in *Example 2: Logging a Temperature Signal with Start and Stop Conditions*.
8. After the start condition is met, the status indicator updates to display **Recording**. If the logged signal does not appear on the **Data View** tab, drag the signal from the Logged Data window to the **Data View** tab to view the progress of the log.
9. When the temperature signal exceeds 85 degrees Fahrenheit, a red alarm indicator appears on the graph of the log. The alarm

remains active until the temperature falls below 85 degrees.

10. The project runs and LabVIEW SignalExpress logs the signal until the stop condition is met. After the stop condition is met, expand the log in the **Logged Data** window, right-click **Thermocouple**, and select **Show Alarms and Events** from the shortcut menu to display a dialog box that list the alarms and events that occurred during the logging operation. You can click the **Export** button on this dialog box to save the list to a text file.
11. Click the **OK** button to close the dialog box and select **File»Save Project** to save the project.

This example and the two that precede it describe different ways to configure logging operations. After a logging operation is complete, you can use LabVIEW SignalExpress to view and analyze the logged signal. Example 4: Analyzing a Logged Temperature Signal describes how to analyze the logged temperature signal in LabVIEW SignalExpress.

Example 4: Analyzing a Logged Temperature Signal

The following procedure describes how to analyze a logged temperature signal to determine the maximum, minimum, and mean values of the signal. This example uses the project you created in Example 1: Logging a Temperature Signal and modified in Example 2: Logging a Temperature Signal with Start and Stop Conditions and Example 3: Displaying Alarms when a Temperature Signal Meets a Specified Value.

Complete the following steps to analyze the logged temperature signal and determine the maximum, minimum, and mean values of the signal.

1. Open the project you saved in *Example 3: Displaying Alarms when a Temperature Signal Meets a Specified Value*.
2. In the work areas pull-down menu that appears above the Project View, select **Playback** to switch to a Playback work area. The Playback work area is similar to the Monitor/Record work area in which you logged the signal except that the Data View tab appears with a time bar and various buttons you can use to navigate a logged signal. You also can use logged signals as step inputs in a Playback work area.
3. In the Logged Data window, right-click the last log you created and select **Make Active Log** from the shortcut menu. The name of the active log appears in bold.
4. Drag the log from the **Logged Data** window to the **Data View** tab.
5. Click the **Run** button to play back the log. You also can drag the slider in the time bar to navigate the log, or you can use the buttons and the pull-down menu on the time bar to adjust how LabVIEW SignalExpress plays back the logged signal. For example, you can adjust the playback speed.
6. Select **Add Step»Analysis»Time-Domain Measurements»Statistics** to add the Statistics step to the Project View. The Step Setup tab for the Statistics step appears, and LabVIEW SignalExpress selects the logged signal as the step input signal automatically.
7. On the **Configuration And Results** page of the **Step Setup** tab, place checkmarks in the **Max**, **Min**, and **Mean** checkboxes to specify that the step returns the maximum, minimum, and mean

values of the temperature signal as outputs.

8. Drag the new step outputs to the **Data View** tab and click the **Run** button to run the project and display the maximum, minimum, and mean values of the temperature signal on the **Data View** tab.
9. Select **File»Save Project** to save the project.

Building VIs for Run LabVIEW VI Steps

You can build a Run LabVIEW VI step from an existing VI or you can start from a template.

Building Run LabVIEW VI Steps from Existing VIs

You can call most VIs from LabVIEW SignalExpress.

Complete the following steps to build a VI you can use in a Run LabVIEW VI step.

Note You must use LabVIEW 7.1 or later to create a VI you can run from the Run LabVIEW VI step. You cannot run a VI that was saved for a previous version of LabVIEW using the Run LabVIEW VI step. You must save the VI in the actual version of LabVIEW you want to run.

1. Open an existing VI to use as a Run LabVIEW VI step. You also can use a template to build a Run LabVIEW VI step.
2. Connect all inputs and outputs necessary for the operation of the VI to the connector pane of the VI. When you call a VI from LabVIEW SignalExpress, it reads the connector pane of the VI to determine the inputs and outputs. If you do not wire the inputs and outputs to the connector pane, LabVIEW SignalExpress cannot pass data into or out of the VI. Connecting the inputs and outputs to the connector pane also enables LabVIEW SignalExpress to properly convert a LabVIEW SignalExpress project to a LabVIEW block diagram. Make sure LabVIEW SignalExpress supports the data types of the controls and indicators.
3. Select **File»VI Properties**, select **Execution** from the **Category** pull-down menu, and place a checkmark in the **Reentrant execution** checkbox. Reentrant VIs create a unique dataspace for each instance of a VI when it is called. Because you can use a single VI multiple times in a LabVIEW SignalExpress project, you must save the VI as reentrant to avoid dataspace clashing.
4. LabVIEW 7.1: Save the VI as an LLB with all of its subVIs included by selecting **File»Save with Options** and clicking the **Application Distribution** option in LabVIEW. When you combine all subVIs into an LLB, you ensure that all of the components necessary to execute the VI are present on the system. Refer to the *LabVIEW Help* for more information about creating linked

libraries in LabVIEW.

LabVIEW 8.0: Save the VI as an source distribution with all of its subVIs included by creating a new project library which includes the VIs. Right-click the **Build Specifications** option in the **Project Explorer** window and select **New»Source Distribution**. In the **Source Distribution Properties** dialog box, remove the checkmarks from the **Exclude vi.lib**, **Exclude instr.lib**, and **Exclude user.lib** options. Click the **Build** button to build the source distribution. Refer to the *LabVIEW Help* for more information about creating project libraries and source distributions in LabVIEW.

LabVIEW 8.2 and later: Save the VI as an source distribution with all of its subVIs included by creating a new project library which includes the VIs. Right-click the **Build Specifications** option in the **Project Explorer** window and select **New»Source Distribution**. On the **Additional Exclusions** page of the **Source Distribution Properties** dialog box, remove the checkmarks from the **Exclude files from vi.lib**, **Exclude files from instr.lib**, and **Exclude files from user.lib** options. Click the **Build** button to build the source distribution. Refer to the *LabVIEW Help* for more information about creating project libraries and source distributions in LabVIEW.

5. Before you use the newly created linked library, close the library and LabVIEW to ensure the VI does not remain in the system memory.

Note Any time a Run LabVIEW VI step uses a dynamic link library (DLL), you must maintain the path to the DLL. If you move the DLL or put the files on a different computer, you must open the Run LabVIEW VI step in LabVIEW and relink the VI to the DLL.

Building Run LabVIEW VI Steps from a Template

Note You must use LabVIEW 7.1 or later to create a VI you can run from the Run LabVIEW VI step. You cannot run a VI that was saved for a previous version of LabVIEW using the Run LabVIEW VI step. You must save the VI in the actual version of LabVIEW you want to run.

The LabVIEW SignalExpress steps have four execution states. In LabVIEW, open UserDefinedStepTemplate.vi in the SignalExpress\User Step Templates directory. The UserDefinedStepTemplate VI contains a Case structure with four cases. Each case represents one of the four execution states. The following list describes each case:

- **Configure**—Executes once as the first case each time you click the **Run** button in LabVIEW SignalExpress.
- **Reconfigure**—Executes when you change parameter values on a Run LabVIEW VI step while the project is running.
- **Run**—Executes repeatedly or once after the **Configure** case executes, depending on the run mode you use to run the project. Place indicators for data you acquire in this case.
- **Stop**—Executes once after you click the **Stop** button or select the **Abort** option in LabVIEW SignalExpress.

Note The _Event enumerated type control on the front panel of the template VI is connected to the connector pane and wired to the selector terminal of the Case structure on the block diagram. Removing or renaming this control breaks the template VI.

If you need to share data between execution cases, use a shift register on the While Loop that surrounds the Case structure. Notice that a Boolean constant wired to the While Loop condition node is set to TRUE. This forces code in the While Loop to execute once every time LabVIEW SignalExpress calls the Run LabVIEW VI step. You only need to use a While Loop if you use shift registers to share data between execution cases.

Calling LabVIEW VIs from LabVIEW SignalExpress

Complete the following steps to use a VI in LabVIEW SignalExpress.

1. Click the **Add Step** button and select the Run LabVIEW VI step for the version of LabVIEW in which you saved your VI.
2. Click the browse button next to the **Select VI** field and navigate to the VI you want to use in the LabVIEW SignalExpress project.
3. Click the **Connect Input** button and select which input on the VI you want to use to pass in a signal from another step in the project. LabVIEW SignalExpress reads the connector pane of the VI to determine the inputs to list in the **Connect Input** dialog box.
4. In the **Input signal** drop-down list, select a signal from a previous step you want to pass into the VI. This drop-down list displays signals for steps that execute before the Run LabVIEW VI step.
5. The Run LabVIEW VI step reads the connector pane of the VI and lists the outputs in the **Outputs** list. For each output signal, LabVIEW SignalExpress selects the appropriate signal type so other steps in the project can use the signals correctly.

After you specify the input and output signals of the Run LabVIEW VI step, you can run the step like any other step within LabVIEW SignalExpress projects. You can view the front panel of the VI and change the parameters of the step while it is running.

Note VIs you use in your Run LabVIEW VI step must be reentrant and must be in LLBs or development distributions.

Converting a Project to a LabVIEW Block Diagram

You can use LabVIEW SignalExpress to build automated measurement projects that generate stimulus and acquire response signals, analyze and display signals, and save the signals. After you create and save a project in LabVIEW SignalExpress, you can convert that project to a LabVIEW block diagram. Converting a LabVIEW SignalExpress project into a LabVIEW block diagram has the following benefits:

- LabVIEW compiles and executes block diagrams faster than LabVIEW SignalExpress projects.
- You can execute LabVIEW block diagrams using TestStand as part of an automated test sequence.
- You can take advantage of LabVIEW to extend your projects in the following ways:
    - Creating a custom user interface with buttons, knobs, and meters to control the execution and display of your measurements.
    - Controlling other measurement hardware not supported by LabVIEW SignalExpress, such as GPIB instruments, distributed I/O devices, such as FieldPoint, machine vision boards, and motion controllers.
    - Performing more advanced analysis routines using the LabVIEW analysis libraries or add-on toolkits.
    - Distributing your project to run or be controlled from multiple computers or across the Web.

Note To convert a LabVIEW SignalExpress project to a LabVIEW VI, you must have the LabVIEW 7.1 Full Development System or later installed.

Complete the following steps to convert a LabVIEW SignalExpress project to a LabVIEW block diagram.

1. Select **Tools»Generate Code»LabVIEW Diagram** to begin the conversion process.
2. Specify a name for the VI you want to generate, and click the **OK** button.

LabVIEW converts the active work area of the LabVIEW SignalExpress project to a LabVIEW VI.

Note If you want to convert a step that contains a parameter set to perform a sweep operation, LabVIEW converts that step to a subVI, not an Express VI. LabVIEW converts other steps within the sweep operation to Express VIs.

When you convert a LabVIEW SignalExpress project with logging, LabVIEW SignalExpress generates a LabVIEW block diagram with one Express VI. You cannot convert the generated Express VI into a subVI. When you double-click the Express VI, LabVIEW opens the LabVIEW SignalExpress project associated with the Express VI. Refer to the KnowledgeBase for more information about the Express VI.

Distributing LabVIEW Block Diagrams for Execution

LabVIEW requires the LabVIEW SignalExpress execution engine to run LabVIEW SignalExpress steps converted to the LabVIEW block diagram. If you distribute the converted VI for use on other computers, make sure the target computer has LabVIEW SignalExpress installed. If the target computer does not have LabVIEW SignalExpress installed, you must use a source distribution to distribute the converted VI. The following sections contains guidelines for distributing VIs.

Distributing a VI to a Computer that Has LabVIEW SignalExpress Installed

Copy the VI to the target computer. You then can run the Express VI, open the Express VI and reconfigure the settings, and convert the Express VI to subVIs if necessary. You might need to update the device settings for any hardware steps you are using to make sure the VI can find the hardware on the target computer correctly.

Distributing a VI to a Computer that Does Not Have LabVIEW SignalExpress Installed

Complete the following steps to distribute a VI to a computer that does not have LabVIEW SignalExpress installed.

1. In LabVIEW, select **File»New Project** to create a new LabVIEW project.
2. In the **Project Explorer** window, right-click **My Computer** and select **Add»File** from the shortcut menu to add the converted VI to the project.
3. Save the project.
4. Right-click **Build Specifications** and select **New»Source Distribution** from the shortcut menu to display the **Source Distribution Properties** dialog box.
5. Enter the location for the source distribution in the **Destination directory** text box. You can use the **Browse** button to navigate to and select a location.
6. Click the **Build** button in the **Source Distribution Properties** dialog box to build the source distribution.

Note If LabVIEW returns an error, click the **Remove unused members of project libraries** option on the **Additional Exclusions** page of the **Source Distribution Properties** dialog box and click the **Build** button.

7. Copy the resulting folder from the destination directory to the target computer where you want to run the VI.
8. Copy the labview\vi.lib\express\SignalExpress\Support folder from the computer that has LabVIEW SignalExpress installed to the labview folder on the target computer.
9. Run the VI on the target computer. If you try to open the Express VI configuration view to reconfigure the operation, LabVIEW displays an error dialog box that indicates that you do not have the LabVIEW SignalExpress execution engine installed.

Note If you need to modify the VI, convert the Express VI into subVIs and modify the subVIs.

Refer to the *LabVIEW Help* for more information about building source distributions in LabVIEW.

Using Express VIs in LabVIEW

When you convert a LabVIEW SignalExpress project to a LabVIEW block diagram, your block diagram contains LabVIEW Express VIs wired together. Typically, each step in LabVIEW SignalExpress corresponds to an Express VI in your LabVIEW block diagram. When you double-click these Express VIs, LabVIEW displays a configuration view that is identical to the configuration view for the corresponding step in LabVIEW SignalExpress. You can reconfigure the execution of your VI in LabVIEW by resetting values in the configuration view. Unlike LabVIEW SignalExpress where you can change the configuration view options while LabVIEW SignalExpress executes, you cannot open the Express VI configuration view and change the settings while LabVIEW executes your VI. You must stop the VI, open the configuration view, make a change, and rerun the VI.

Note Because the LabVIEW SignalExpress execution engine manages the configuration views for the Express VIs running in LabVIEW, you must have LabVIEW SignalExpress installed on the same computer that you are running LabVIEW in order for these Express VIs to function properly. If you do not have LabVIEW SignalExpress installed on the computer, you must convert the Express VIs into subVIs if you want to change their configurations.

Like most Express VIs, you can convert LabVIEW SignalExpress Express VIs to LabVIEW subVIs. When you convert LabVIEW SignalExpress projects into LabVIEW block diagrams, you might need to modify the low-level VIs rather than reconfigure the values in the Express VIs. To convert the Express VIs into subVIs to access the low-level VIs, right-click the Express VI, select **Open Front Panel**, and click the **Convert** button.

Using the LabVIEW SignalExpress Express VIs with Native LabVIEW Express VIs

You can build VIs using the **LabVIEW SignalExpress** palette in LabVIEW. These Express VIs use the LabVIEW waveform data type, not the dynamic data type the native LabVIEW Express VIs use. The dynamic data type represents an array of waveforms.

In some cases, LabVIEW cannot convert the LabVIEW SignalExpress project into Express VIs because some Express VIs might not support the functionality your project uses. The following examples describe the functionality not supported by Express VIs:

- Hardware synchronization—LabVIEW converts any project that uses the synchronization features of the measurement hardware, such as trigger sharing, clock sharing, and so on, into subVIs rather than Express VIs. The measurement hardware Express VIs for NI digitizers, arbitrary waveform or function generators, and multifunction DAQ boards do not support these synchronization features. In addition, you can preserve the dependency between a generator and a measurement device when performing a stimulus or response measurement in a single subVI that contains code for both the generator and the measurement device together.
- Sweeping—LabVIEW converts the Sweep step into a For Loop in LabVIEW. The For Loop generates new values for each iteration and passes these values to the VIs that accept these values as inputs. Because Express VIs cannot accept new values on a wire during execution, LabVIEW implements sweeping in subVIs.
- Logging—When you convert a LabVIEW SignalExpress project with logging, LabVIEW SignalExpress generates a LabVIEW block diagram with one Express VI. You cannot convert the generated Express VI into a subVI.

Running and Modifying Converted Projects in LabVIEW

When you convert a LabVIEW SignalExpress project to a LabVIEW block diagram, the resulting LabVIEW block diagram represents the exact functionality of the LabVIEW SignalExpress project.

However, when you convert a LabVIEW SignalExpress project to a LabVIEW block diagram, the front panel of the VI that you generate contains only the controls that are necessary for execution and indicators that match the data types of the output signals from the converted project. Any graph, chart, or other displays that the LabVIEW SignalExpress project contains are not replicated on the LabVIEW front panel. However, you can use LabVIEW controls and indicators to build a front panel for the VI and define a custom user interface. For example, you can create graphs on which to display the output signals.

Refer to the *LabVIEW Help* for more information about building front panels in LabVIEW.

Step Reference

This section contains the steps you can use to build measurement tasks.

LabVIEW SignalExpress Steps

Use the LabVIEW SignalExpress steps to build interactive measurement applications.

 To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Acquire Signals

Use the Acquire Signals steps to acquire signals from a hardware device.

 To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

IVI Scope Acquire

Acquires an analog waveform from an instrument in the Oscilloscope IVI Class.

Default values are specific to the hardware and driver specified. The default settings might not be applicable to the measurement you are trying to perform.

The IVI Scope Acquire toolbar includes two buttons you can use to set parameter values. Click the **Initialize** button to set IVI Scope Acquire to the default settings. Click the **Autosetup** button to set parameters to values that IVI Scope Acquire determines best fit the signal you are acquiring.

Note Clicking the **Autosetup** button executes IVI Scope Acquire.

To communicate with an instrument, you need to install the instrument-specific driver and create a session name for the instrument.

Parameter	Description
Autoscale amplitude	Scales the amplitude axis of the **Acquired signals** graph. The default is to autoscale the amplitude.
Acquired Signals	Displays the waveform from the device. **Range (V)** and **Offset (V)** set the hardware limits.
Configuration	Contains the following configuration options: • **Device**—Contains the following device options: – **IVI session name**—Specifies the session name to use for this step. This step retrieves possible session names from National Instruments Measurement & Automation Explorer (MAX). You also can create a new session or edit/delete an existing session. – **Resource descriptor**—Specifies the interface and the address of the hardware to associate with the step. – **Instrument driver**—Displays the name of the driver in use.

- **Vertical**—Contains channel configuration options that affect the data along the Voltage (V) axis. The settings you configure with these options are specific to the channel you select in the **Channels** field. **Vertical** contains the following options:
  - **Channels**—Specifies the physical channels from which to generate data.
  - **Enable channel**—Specifies whether to enable data acquisition on the selected channel.
  - **Range (V)**—Specifies the value of the input range the oscilloscope uses for the channel. For example, to acquire a sine wave that spans -5 to 5 volts, enter 10 as the value of this parameter.
  - **Input impedance (Ohms)**—Specifies the input impedance you want to use for the channel.
  - **Probe attenuation**—Specifies the scaling factor by which the probe you attach to the channel attenuates the input. Pass -1 to auto detect.
  - **Offset (V)**—Specifies the location of the center of the range that you specify with **Range (V)**. Enter the value with respect to ground. For example, to acquire a sine wave that spans 0 to 10 volts, enter 5 as the value of this parameter.
  - **Coupling**—Specifies how you want the oscilloscope to couple the input signal for the channel. Options include AC, DC, and GND.
  - **Bandwidth (Hz)**—Specifies the maximum frequency for the input signal you want the instrument to accommodate without attenuating the

	signal by more than 3 dB. • **Horizontal**—Contains the following device-specific options for configuring the Time (s) axis: – **Start time (s)**—Specifies the length of time from the trigger event to the first point in the waveform record. If this value is positive, the first point in the waveform record occurs after the trigger event. If this value is negative, the first point in the waveform record occurs before the trigger event. – **Time per record (s)**—Specifies the time in seconds that corresponds to the record length. – **Min record length (S)**—Specifies the minimum number of points you require in the waveform record for each channel.
Trigger	Contains the following trigger options: • **Type**—Specifies the type of trigger you want the oscilloscope to use. Contains the following options: – **Immediate**— Configures the oscilloscope for immediate triggering. The oscilloscope does not wait for a trigger of any kind upon initialization. – **Edge**—Configures the oscilloscope for edge triggering. An edge trigger occurs when the trigger signal crosses the trigger level you specify with the slope you specify. – **TV**—Configures the oscilloscope for TV triggering. – **Runt**—Configures the oscilloscope for runt triggering. A runt trigger occurs when the trigger signal crosses one of the runt thresholds twice without crossing the other runt threshold.

- **Glitch**—Configures the oscilloscope for glitch triggering. A glitch trigger occurs when the trigger signal has a pulse with a width that is less than the glitch width. The trigger does not actually occur until the edge of the pulse that corresponds to the glitch width and polarity you specify crosses the trigger level.
- **Width**—Configures the oscilloscope for width triggering. A width trigger occurs when the oscilloscope detects a positive or negative pulse with a width between, or optionally outside, the width thresholds. The trigger does not actually occur until the edge of a pulse that corresponds to the width thresholds and polarity you specify crosses the trigger level.
- **AC Line**—Configures the oscilloscope for AC line triggering.

- **Source**—Specifies the source for the oscilloscope to monitor for a trigger.
- **Holdoff (s)**—Specifies the length of time you want the oscilloscope to wait after it detects a trigger until the oscilloscope enables the trigger subsystem to detect another trigger.
- **Timeout (s)**—Specifies the maximum amount of time to wait for the oscilloscope to acquire data. When a timeout occurs during an acquisition, it is normally due to a failure to trigger. The default is 10.
- **Level (V)**—[Type: Edge] Specifies the voltage you want the oscilloscope to use for edge triggering. The oscilloscope triggers when the trigger signal passes through the threshold you specify with this parameter and has the slope you specify with the **Slope** parameter.

Note This parameter affects instrument behavior only when you select a channel or the external trigger input as the trigger source. You may not configure the trigger level that the oscilloscope uses for other trigger sources, such as VXI TTL trigger lines.

- **Slope**—[Type: Edge] Specifies whether you want a rising edge or a falling edge passing through the trigger level to trigger the oscilloscope. Options include Positive and Negative.
- **Coupling**—[Type: Edge] Specifies the trigger coupling. Options include AC, DC, HF Reject, LF Reject, and Noise Reject.
- **Polarity**—[Type: TV] Specifies the polarity of the TV signal. Options include Positive and Negative.
- **Signal format**—[Type: TV] Specifies the type of TV signal on which the oscilloscope triggers. Options include NTSC, PAL, and SECAM.
- **Event**—[Type: TV] Specifies the TV event on which you want the oscilloscope to trigger. Options include Field 1, Field 2, Any Field, Any Line, and Line Number.
- **Line number**—[Type: TV] Specifies the line in the field on which you want the oscilloscope to trigger. The specified line number is independent of any field. This means that to trigger on the first line of **Field 2**, you must specify a line number of 263 (if we assume that **Field 1** has 262 lines).
- **Polarity**—[Type: Runt] Specifies the polarity of the runt that you want to trigger the oscilloscope. Contains the following options:
  - **Positive**—Triggers on a positive runt. A positive runt occurs when a rising edge crosses the **Low threshold (V)** and does

not cross the **High threshold (V)** before recrossing the **Low threshold (V)**.

  - **Negative**—Triggers on a negative runt. A negative runt occurs when a falling edge crosses the **High threshold (V)** and does not cross the **Low threshold (V)** before recrossing the **High threshold (V)**.
  - **Either**—Triggers on either a positive or negative runt.

- **Low threshold (V)**—[Type: Runt] Specifies the low threshold you want the oscilloscope to use for runt triggering.
- **High threshold (V)**—[Type: Runt] Specifies the high threshold you want the oscilloscope to use for runt triggering.
- **Level (V)**—[Type: Glitch] Specifies the voltage threshold you want the oscilloscope to use for glitch triggering. The oscilloscope triggers when a glitch crosses the trigger threshold you specify with this parameter.
- **Polarity**—[Type: Glitch] Specifies the polarity of the glitch that you want to trigger the oscilloscope. Options include Positive, Negative, and Either.
- **Condition**—[Type: Glitch] Specifies the glitch condition. The oscilloscope triggers when it detects a pulse with a width less than or greater than the **Width (s)** value. Options include Less Than and Greater Than.
- **Width (s)**—[Type: Glitch] Specifies the length of time you want the oscilloscope to use for the glitch width. The oscilloscope triggers when it detects a pulse with a width less than or greater than this value, depending on the **Condition** parameter.
- **Level (V)**—[Type: Width] Specifies the voltage

<table>
<tr><td></td><td>threshold you want the oscilloscope to use for width triggering. The oscilloscope triggers when the edge of a pulse that corresponds to the Low threshold (V), High threshold (V), Condition, and Polarity crosses the threshold you specify in this parameter.

Polarity—[Type: Width] Specifies the polarity of the pulse that you want to trigger the oscilloscope. Options include Positive and Negative.
High threshold (V)—[Type: Width] Specifies the high width threshold.
Low threshold (V)—[Type: Width] Specifies the low width threshold.
Condition—[Type: Width] Specifies whether you want a pulse that is within or outside the High threshold (V) and Low threshold (V) to trigger the oscilloscope. Contains the following options:

Within—Triggers on pulses that have a width that is less than the High threshold (V) and greater than the Low Threshold (V).
Outside—Triggers on pulses that have a width that is either greater than the High threshold (V) or less than the Low threshold (V).

Slope—[Type: AC Line] Specifies whether you want the oscilloscope to trigger on a zero crossing with a positive, negative, or either slope of the network supply voltage. Options include Positive, Negative, and Either.
</td></tr>
<tr><td>Advanced</td><td>Contains the following option:

Acquisition Settings—Contains the following option:

Acquisition type—Specifies the manner in which you want the oscilloscope to acquire data and fill the waveform

</td></tr>
</table>

		record. Contains the following options: • **Normal**—Sets the oscilloscope to normal acquisition mode. The oscilloscope acquires one sample for each point in the waveform record. The oscilloscope can use real-time or equivalent-time sampling. • **Peak Detect**—Sets the oscilloscope to the peak-detect acquisition mode. The oscilloscope oversamples the input signal and keeps the minimum and maximum values that correspond to each position in the waveform record. The oscilloscope uses only real-time sampling. • **High Resolution**—Sets the oscilloscope to the high-resolution acquisition mode. The oscilloscope oversamples the input signal and calculates an average value for each position in the waveform record. The oscilloscope uses only real-time sampling. • **Envelope**—Sets the oscilloscope to the envelope acquisition mode. The oscilloscope acquires multiple waveforms and keeps the minimum and maximum voltages it acquires for each point in the waveform record. The oscilloscope can use real-time or equivalent-time sampling. • **Average**—Sets the oscilloscope

<table>
<tr>
<td></td>
<td>to the average acquisition mode. The oscilloscope acquires multiple waveforms and calculates an average value for each point in the waveform record. The oscilloscope can use real-time or equivalent-time sampling.

Note When you set this parameter to Envelope or Peak Detect, the oscilloscope acquires minimum and maximum waveforms.</td>
</tr>
<tr>
<td>Execution Control</td>
<td>Contains the following execution control options:

Start this step after—Makes the step wait until another step has started before executing. You can make the step wait on any other hardware step in the project by selecting the step to wait on from the pull-down menu.

You can use this option to force an acquisition device to start after a generation device starts. You also can use this option to ensure that a device generating a trigger signal starts after the device receiving the signal, which avoids sending the signal before the receiver is ready.
Step to wait for—Lists the possible steps for which this step can wait.
Pre-execution delay (ms)—Specifies the amount of time to wait before the step executes. If you configure the step to start after another step, the delay represents the amount of time to wait after the specified step starts.
Post-execution delay (ms)—Specifies the amount of time to wait after the step executes.

</td>
</tr>
</table>

IVI DMM Acquire

Acquires a signal from an instrument in the Digital Multimeter IVI Class.

Default values are specific to the hardware and driver specified. The default settings may not be applicable to the measurement you are trying to perform. Click the **Initialize** button, located on the step's toolbar, at any time to reset the step to the default settings.

To communicate with an instrument, you need to install the instrument-specific driver and create a session name for the instrument.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Output Display</td><td>Displays the measurement, formatted according to the Measurement function, Range, and Resolution.</td></tr>
<tr><td>Configuration</td><td>Contains the following configuration options:

Device—Contains the following device options:

IVI session name—Specifies the session name to use for this step. This step retrieves possible session names from National Instruments Measurement & Automation Explorer (MAX). You also can create a new session or edit/delete an existing session.
Resource descriptor—Specifies the interface and the address of the hardware to associate with the step.
Instrument driver—Displays the name of the driver in use.

Basic Parameters—Contains the following options:

Measurement function—Specifies the type of measurement you want the DMM to perform. Options include DC Volts, AC Volts, DC Current, AC Current, 2 Wire Resistance, 4 Wire Resistance, AC + DC Volts, AC + DC Current, Frequency, and Period.
Range—Specifies whether Auto Range

</td></tr>
</table>

is used. Contains the following options:

- **Auto Range**—Specifies that the DMM automatically calculates the range before each measurement.
- **Specify Range**—Allows you to specify the range and uses this value for all subsequent measurements until you change the measurement configuration.

– **Range value (V)**—[Measurement function: DC Volts, AC Volts, AC + DC Volts] The range in volts for the current measurement.
– **Range value (A)**—[Measurement function: DC Current, AC Current, AC + DC Current] The range in amps for the current measurement.
– **Range value (Ohm)**—[Measurement function: 2 Wire Resistance, 4 Wire Resistance] The range in ohms for the current measurement.
– **Range value (Hz)**—[Measurement function: Frequency] The range in hertz for the current measurement.
– **Range value (s)**—[Measurement function: Period] The range in seconds for the current measurement.
– **Resolution**—Specifies the digital resolution of the measurement. Set **Range** to **Specify Range** to enable this option.
– **Sample period (s)**—Specifies how often to execute the step.

- **Measurement Specific Parameters**—Contains the following options:
  – **Auto zero**—Specifies that the DMM

internally disconnects the input signal and takes a zero reading. The DMM then subtracts the zero reading from the measurement to prevent offset voltages present from affecting measurement accuracy. This option does not appear if you set **Measurement function** to **Frequency** or **Period**. Contains the following options:

- **On**—Configures the DMM to take a zero reading for each measurement. The DMM subtracts the zero reading from the value it measures.
- **Off**—Disables the **Auto zero** option.
- **Once**—Configures the DMM to take a zero reading immediately. The DMM then subtracts this zero reading from all subsequent values it measures.

- **AC min frequency (Hz)**—[Measurement function: AC Volts, AC Current, AC + DC Volts, AC + DC Current] Specifies the minimum expected frequency component of the input signal in hertz.
- **AC max frequency (Hz)**—[Measurement function: AC Volts, AC Current, AC + DC Volts, AC + DC Current] Specifies the maximum expected frequency component of the input signal in hertz.
- **Frequency voltage range**—[Measurement function: Frequency, Period] Specifies whether the frequency voltage **Auto Range** is used. Contains the following options:

	Auto Range—Configures the DMM to automatically calculate the voltage range before each frequency or period measurement. **Specify Range**—Disables auto ranging. The DMM sets the voltage range to the range specified in **Frequency range (V)**. **Frequency range (V)**—[Measurement function: Frequency, Period] Specifies the expected maximum amplitude of the input signal. The minimum peak-to-peak signal amplitude that the DMM can detect is 10% of the specified voltage range. **Powerline Frequency (Hz)**—Specifies the powerline frequency in hertz.
Trigger	Contains the following trigger options: **Type**—Specifies the trigger source you want to use. After the DMM receives the trigger, the DMM waits the length of time you specify in the **Delay (s)** parameter. The DMM then takes a measurement. The default is Immediate. Contains the following options: **Immediate**—Does not wait for a trigger of any kind. **External**—Waits for a trigger on the external input. **Software**—Waits until you press the associated trigger button in the toolbar. **Digital**—Waits for a trigger on a digital input specified by the trigger source. **Delay (s)**—Specifies the length of time the DMM waits after it receives the trigger and before it takes a measurement.

- **Auto**—Configures the DMM to automatically calculate the trigger delay before each measurement.
- **Timeout (s)**—Configures the amount of time to wait while retrieving a reading from the DMM. The default is 5 seconds.
- **Slope**—[Type: External] Specifies whether you want a rising edge or a falling edge passing through the trigger level to trigger the DMM. Contains the following options:
  - **Positive**—Triggers on the rising edge of the external trigger.
  - **Negative**—Triggers on the falling edge of the external trigger.
- **Software trigger source**—[Type: Software] Specifies the trigger source to which you want the instrument to respond. To activate the trigger, click the associated trigger button in the toolbar. Execution waits until you click the associated trigger button in the toolbar. Contains the following options:
  - **Trigger A**—(Default) Specifies Trigger A as the trigger source.
  - **Trigger B**—Specifies Trigger B as the trigger source.
  - **Trigger C**—Specifies Trigger C as the trigger source.
- **Source**—[Type: Digital] Specifies the trigger source you want to use. Contains the following options:
  - **PXI TRIG0 or VXI TTL0**—Waits until it receives a trigger on the PXI TRIG0 line (for PXI instruments) or the VXI TTL0 line (for VXI instruments).
  - **PXI TRIG1 or VXI TTL1**—Waits until it receives a trigger on the PXI TRIG1 line (for PXI instruments) or the VXI TTL1

		line (for VXI instruments). – **PXI TRIG2 or VXI TTL2**—Waits until it receives a trigger on the PXI TRIG2 line (for PXI instruments) or the VXI TTL2 line (for VXI instruments). – **PXI TRIG3 or VXI TTL3**—Waits until it receives a trigger on the PXI TRIG3 line (for PXI instruments) or the VXI TTL3 line (for VXI instruments). – **PXI TRIG4 or VXI TTL4**—Waits until it receives a trigger on the PXI TRIG4 line (for PXI instruments) or the VXI TTL4 line (for VXI instruments). – **PXI TRIG5 or VXI TTL5**—Waits until it receives a trigger on the PXI TRIG5 line (for PXI instruments) or the VXI TTL5 line (for VXI instruments). – **PXI TRIG6 or VXI TTL6**—Waits until it receives a trigger on the PXI TRIG6 line (for PXI instruments) or the VXI TTL6 line (for VXI instruments). – **PXI TRIG7 or VXI TTL7**—Waits until it receives a trigger on the PXI TRIG7 line (for PXI instruments) or the VXI TTL7 line (for VXI instruments). – **ECL0**—Waits until it receives a trigger on the VXI ECL0 line. – **ECL1**—Waits until it receives a trigger on the VXI ECL1 line. – **PXI Star**—Waits until it receives a trigger on the PXI STAR trigger bus. – **RTSI 0**—Waits until it receives a trigger on RTSI line 0. – **RTSI 1**—Waits until it receives a trigger on RTSI line 1. – **RTSI 2**—Waits until it receives a trigger on RTSI line 2.

	– **RTSI 3**—Waits until it receives a trigger on RTSI line 3. – **RTSI 4**—Waits until it receives a trigger on RTSI line 4. – **RTSI 5**—Waits until it receives a trigger on RTSI line 5. – **RTSI 6**—Waits until it receives a trigger on RTSI line 6.
Execution Control	Contains the following execution control options: • **Start this step after**—Makes the step wait until another step has started before executing. You can make the step wait on any other hardware step in the project by selecting the step to wait on from the pull-down menu. You can use this option to force an acquisition device to start after a generation device starts. You also can use this option to ensure that a device generating a trigger signal starts after the device receiving the signal, which avoids sending the signal before the receiver is ready. • **Step to wait for**—Lists the possible steps for which this step can wait. • **Pre-execution delay (ms)**—Specifies the amount of time to wait before the step executes. If you configure the step to start after another step, the delay represents the amount of time to wait after the specified step starts. • **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes.

Read Shared Variables

Reads the values of shared variables created in LabVIEW SignalExpress and LabVIEW, as well as data published using DataSocket technology or data that meets OPC specifications. To select data or a shared variable to read, click the **Browse** button to display the Select Network Item dialog box and navigate to the data or shared variable. You also can add machines to the list that appears in the **Select Network Item** dialog box to search for additional data or shared variables.

Parameter	Description
Step Configuration	Contains the following option: • **Sample period (s)**—Specifies the period (in seconds) at which to read data.
Add Shared Variable	Contains the following options: • **Network path**—Specifies the path to the data or shared variable to read. • **Browse**—Opens the **Select Network Item** dialog box, which allows you to browse to the network location of data or a shared variable. • **Add**—Adds the data or shared variable specified in the **Network path** to the step.
Network Paths	Displays the network paths of the data or shared variables LabVIEW SignalExpress is reading.
Remove path	Removes the selected data or shared variable from the step.

Generate Signals

Use the Generate Signals steps to generate signals to a hardware device.

 To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

IVI FGEN Standard Function

Generates an analog standard function using an instrument in the Arbitrary Waveform/Function Generator IVI class.

Default values are specific to the hardware and driver specified. The default settings may not be applicable to the measurement you are trying to perform. Click the **Initialize** button, located on the step's toolbar, at any time to reset the step to the default settings.

To communicate with an instrument, you need to install the instrument-specific driver and create a session name for the instrument.

Parameter	Description
Function Preview	Displays a preview of the function. The vertical and horizontal graph axes are formatted according to the **Channel Configuration** settings that are applied to the signal generator.
Configuration	Contains the following configuration options: • **Device**—Contains the following device options: – **IVI session name**—Specifies the session name to use for this step. This step retrieves possible session names from National Instruments Measurement & Automation Explorer (MAX). You also can create a new session or edit/delete an existing session. – **Resource descriptor**—Specifies the interface and the address of the hardware to associate with the step. – **Instrument driver**—Displays the name of the driver in use. • **Channel Configuration**—Contains the following channel configuration options: – **Channels**—Specifies the physical channels on which data is generated. – **Enable channel**—Specifies whether to enable data acquisition on the selected channel.

- **Type**—Specifies the standard waveform that you want the function generator to produce. Options include Sine, Square, Triangle, Ramp Up, Ramp Down, and DC.
- **Amplitude (Vpp)**—Specifies the amplitude of the standard waveform that you want the function generator to produce. This value is the amplitude at the output terminal. For example, to produce a waveform ranging from -5 to +5 volts, set the **Amplitude (Vpp)** to 10 volts.
- **Start phase (deg)**—Specifies the horizontal offset of the standard waveform you want the function generator to produce. You specify this property in degrees of one waveform cycle. A start phase of 180 degrees means output generation begins halfway through the waveform. A start phase of 360 degrees offsets the output by an entire waveform cycle, which is identical to a start phase of 0 degrees.
- **Frequency (Hz)**—Specifies the frequency of the standard waveform that you want the function generator to produce.
- **DC offset (V)**—Specifies the DC offset of the standard waveform that you want the function generator to produce. The value is the offset from ground to the center of the waveform you specify with the **Type** parameter. For example, to configure a waveform with an amplitude of 10 volts to range from 0 to +10 volts, set **DC offset (V)** to 5 volts.
- **Output impedance (Ohms)**—Specifies

<table>
<tr><td></td><td>

the impedance value you want the function generator to use. A value of 0 indicates that the function generator is connected to a high impedance load.
Duty cycle (%)—Specifies the percentage of time a square wave remains high versus one entire period. The default is 50%. Duty cycle (%) is available only when you select Square in Type.

Generation Mode—Contains the following generation mode options:

Generate continuously—Generates the input signal continuously. If you run the project continuously, the step generates the input signal repeatedly without discontinuities. If you run the project in Run Once mode, the step generates the input signal once.
Generate N waveforms—Generates the input signal N times in a non-continuous fashion. If you run the project continuously, the step generates the input signal repeatedly but discontinuously. If you run the project in Run Once mode, the step generates the input signal once. You can use this option if you want the device to generate a start trigger every time the device starts generating the signal.
Number of waveforms—Specifies the number of times to generate the waveform.

</td></tr>
<tr><td>Trigger</td><td>Contains the following trigger options:
Channel Triggering—Contains the following channel trigger options:
Channels—Specifies the physical

</td></tr>
</table>

channels that have a trigger associated with them. Each channel can be triggered independently.

- **Type**—Specifies the trigger source to which you want the function generator to respond. The default is Immediate. Contains the following options:
    - **Immediate**—Does not wait for a trigger of any kind.
    - **Internal**—Waits for a trigger on the internal trigger input.
    - **External**—Waits for a trigger on the external trigger input.
    - **Software**—Waits until the software trigger button specified by the **Software trigger source** is pressed on the toolbar.
    - **Digital**—Waits for a trigger on a digital input specified by the trigger source.
- **Internal trigger rate (trig/s)**—Specifies the rate at which you want the internal trigger rate of the function generator to generate trigger signals. **Internal trigger rate (trig/s)** is available only when you select **Internal** in **Type**.
- **Software trigger source**—[Type: Software] Specifies the trigger source to which you want the instrument to respond. To activate the trigger, click the associated trigger button in the toolbar. Execution waits until you click the associated trigger button in the toolbar. Contains the following options:
    - **Trigger A**—(Default) Specifies Trigger A as the trigger source.
    - **Trigger B**—Specifies Trigger B

as the trigger source.

- **Trigger C**—Specifies Trigger C as the trigger source.

– **Source**—[Type: Digital] Specifies the trigger source you want to use. Contains the following options:

  - **PXI TRIG0 or VXI TTL0**—Waits until it receives a trigger on the PXI TRIG0 line (for PXI instruments) or the VXI TTL0 line (for VXI instruments).
  - **PXI TRIG1 or VXI TTL1**—Waits until it receives a trigger on the PXI TRIG1 line (for PXI instruments) or the VXI TTL1 line (for VXI instruments).
  - **PXI TRIG2 or VXI TTL2**—Waits until it receives a trigger on the PXI TRIG2 line (for PXI instruments) or the VXI TTL2 line (for VXI instruments).
  - **PXI TRIG3 or VXI TTL3**—Waits until it receives a trigger on the PXI TRIG3 line (for PXI instruments) or the VXI TTL3 line (for VXI instruments).
  - **PXI TRIG4 or VXI TTL4**—Waits until it receives a trigger on the PXI TRIG4 line (for PXI instruments) or the VXI TTL4 line (for VXI instruments).
  - **PXI TRIG5 or VXI TTL5**—Waits until it receives a trigger on the PXI TRIG5 line (for PXI instruments) or the VXI TTL5 line (for VXI instruments).
  - **PXI TRIG6 or VXI TTL6**—Waits

until it receives a trigger on the PXI TRIG6 line (for PXI instruments) or the VXI TTL6 line (for VXI instruments).

- **PXI TRIG7 or VXI TTL7**—Waits until it receives a trigger on the PXI TRIG7 line (for PXI instruments) or the VXI TTL7 line (for VXI instruments).
- **ECL0**—Waits until it receives a trigger on the VXI ECL0 line.
- **ECL1**—Waits until it receives a trigger on the VXI ECL1 line.
- **PXI Star**—Waits until it receives a trigger on the PXI STAR trigger bus.
- **RTSI 0**—Waits until it receives a trigger on RTSI line 0.
- **RTSI 1**—Waits until it receives a trigger on RTSI line 1.
- **RTSI 2**—Waits until it receives a trigger on RTSI line 2.
- **RTSI 3**—Waits until it receives a trigger on RTSI line 3.
- **RTSI 4**—Waits until it receives a trigger on RTSI line 4.
- **RTSI 5**—Waits until it receives a trigger on RTSI line 5.
- **RTSI 6**—Waits until it receives a trigger on RTSI line 6.

- **Reference clock source**—Specifies the reference clock source you want the function generator to use. The function generator derives the frequencies and sample rates that it uses to generate waveforms from the source you specify. Options include Internal, External, and RTSI Clock. For example, when you set

	Reference clock source to **External**, the function generator uses the signal it receives at its external clock terminal as its reference clock.
Execution Control	Contains the following execution control options: • **Start this step after**—Makes the step wait until another step has started before executing. You can make the step wait on any other hardware step in the project by selecting the step to wait on from the pull-down menu. You can use this option to force an acquisition device to start after a generation device starts. You also can use this option to ensure that a device generating a trigger signal starts after the device receiving the signal, which avoids sending the signal before the receiver is ready. • **Step to wait for**—Lists the possible steps for which this step can wait. • **Pre-execution delay (ms)**—Specifies the amount of time to wait before the step executes. If you configure the step to start after another step, the delay represents the amount of time to wait after the specified step starts. • **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes.

IVI FGEN Arbitrary Waveform

Generates an analog arbitrary signal using an instrument in the Arbitrary Waveform/Function Generator IVI class.

Default values are specific to the hardware and driver specified. The default settings may not be applicable to the measurement you are trying to perform. Click the **Initialize** button, located on the step's toolbar, at any time to reset the step to the default settings.

To communicate with an instrument, you need to install the instrument-specific driver and create a session name for the instrument.

Parameter	Description
Waveform Preview	Displays a preview of the waveform downloaded to the function generator. The vertical and horizontal graph axes are formatted according to the **Sample rate (S/s)**, **Gain**, and **Offset (V)** settings applied to the function generator.
Configuration	Contains the following arbitrary waveform configuration options: • **Device**—Contains the following device options: – **IVI session name**—Specifies the session name to use for this step. This step retrieves possible session names from National Instruments Measurement & Automation Explorer (MAX). You also can create a new session or edit/delete an existing session. – **Resource descriptor**—Specifies the interface and the address of the hardware to associate with the step. – **Instrument driver**—Displays the name of the driver in use. • **Channel Configuration**—Contains the following channel configuration options: – **Channels**—Specifies the physical channels on which data is generated. – **Enable channel**—Specifies whether to

		enable data acquisition on the selected channel. – **Input signal**—Select the appropriate waveform to generate. – **Generation Mode**—Contains the following generation mode options: • **Generate continuously**—Generates the input signal continuously. If you run the project continuously, the step generates the input signal repeatedly without discontinuities. If you run the project in Run Once mode, the step generates the input signal once. • **Generate N waveforms**—Generates the input signal N times in a non-continuous fashion. If you run the project continuously, the step generates the input signal repeatedly but discontinuously. If you run the project in Run Once mode, the step generates the input signal once. You can use this option if you want the device to generate a start trigger every time the device starts generating the signal. • **Number of waveforms**—Number of times to generate the **Input Signal**. – **Output Signal**—Contains the following output signal options: • **Extract from waveform**—Specifies whether settings for

Sample rate (S/s), **Gain**, and **Offset (V)** are extracted from the waveform or specified manually. When you remove the checkmark from the **Extract from waveform** checkbox, you first must normalize the data points to a range of -1 to +1.

- **Sample rate (S/s)**—Specifies the sample rate at which you want the function generator to output arbitrary waveforms.
- **Gain**—Specifies the factor by which the function generator scales the arbitrary waveform data. When you create arbitrary waveforms, you first must normalize the data points to a range of -1 to +1. You use this property to scale the arbitrary waveform to other ranges. For example, to configure the output signal to range from -2 to +2 volts, set **Gain** to 2.
- **Offset (V)**—Specifies the value the function generator adds to the arbitrary waveform data. When you create arbitrary waveforms, you first must normalize the data points to a range of -1 to +1. You use this parameter to shift the range of the arbitrary waveform. For example, to configure the output signal to range from 0 to 2 volts instead of -1 to 1 volts, set **Offset (V)** to 1.
- **Impedance (Ohms)**—Specifies

		the impedance value you want the function generator to use. A value of 0 indicates that the function generator is connected to a high impedance load. • **Frequency (Hz)**—Specifies the frequency at which you want the function generator to produce one cycle of an arbitrary waveform.
Trigger	Contains the following trigger options: • **Channel Triggering**—Contains the following channel trigger options: – **Channels**—Specifies the physical channels that have a trigger associated with them. Each channel can be triggered independently. – **Type**—Specifies the trigger source to which you want the function generator to respond. The default is Immediate. Contains the following options: • **Immediate**—Does not wait for a trigger of any kind. • **Internal**—Waits for a trigger on the internal trigger input. • **External**—Waits for a trigger on the external trigger input. • **Software Trigger Function**—Waits until the software trigger button specified by the **Software trigger source** is pressed on the toolbar. • **Digital**—Waits for a trigger on a digital input specified by the trigger source. – **Internal trigger rate (trig/s)**—Specifies	

the rate at which you want the internal trigger rate of the function generator to generate trigger signals. **Internal trigger rate (trig/s)** is available only when you select **Internal** in **Type**.

- **Software trigger source**—[Type: Software] Specifies the trigger source to which you want the instrument to respond. To activate the trigger, click the associated trigger button in the toolbar. Execution waits until you click the associated trigger button in the toolbar. Contains the following options:
    - **Trigger A**—(Default) Specifies Trigger A as the trigger source.
    - **Trigger B**—Specifies Trigger B as the trigger source.
    - **Trigger C**—Specifies Trigger C as the trigger source.
- **Source**—[Type: Digital] Specifies the trigger source you want to use. Contains the following options:
    - **PXI TRIG0 or VXI TTL0**—Waits until it receives a trigger on the PXI TRIG0 line (for PXI instruments) or the VXI TTL0 line (for VXI instruments).
    - **PXI TRIG1 or VXI TTL1**—Waits until it receives a trigger on the PXI TRIG1 line (for PXI instruments) or the VXI TTL1 line (for VXI instruments).
    - **PXI TRIG2 or VXI TTL2**—Waits until it receives a trigger on the PXI TRIG2 line (for PXI instruments) or the VXI TTL2 line (for VXI instruments).

		- **PXI TRIG3 or VXI TTL3**—Waits until it receives a trigger on the PXI TRIG3 line (for PXI instruments) or the VXI TTL3 line (for VXI instruments). - **PXI TRIG4 or VXI TTL4**—Waits until it receives a trigger on the PXI TRIG4 line (for PXI instruments) or the VXI TTL4 line (for VXI instruments). - **PXI TRIG5 or VXI TTL5**—Waits until it receives a trigger on the PXI TRIG5 line (for PXI instruments) or the VXI TTL5 line (for VXI instruments). - **PXI TRIG6 or VXI TTL6**—Waits until it receives a trigger on the PXI TRIG6 line (for PXI instruments) or the VXI TTL6 line (for VXI instruments). - **PXI TRIG7 or VXI TTL7**—Waits until it receives a trigger on the PXI TRIG7 line (for PXI instruments) or the VXI TTL7 line (for VXI instruments). - **ECL0**—Waits until it receives a trigger on the VXI ECL0 line. - **ECL1**—Waits until it receives a trigger on the VXI ECL1 line. - **PXI Star**—Waits until it receives a trigger on the PXI STAR trigger bus. - **RTSI 0**—Waits until it receives a trigger on RTSI line 0. - **RTSI 1**—Waits until it receives a trigger on RTSI line 1. - **RTSI 2**—Waits until it receives a

<table>
<tr><td></td><td>

trigger on RTSI line 2.
RTSI 3—Waits until it receives a trigger on RTSI line 3.
RTSI 4—Waits until it receives a trigger on RTSI line 4.
RTSI 5—Waits until it receives a trigger on RTSI line 5.
RTSI 6—Waits until it receives a trigger on RTSI line 6.

Reference clock source—Specifies the reference clock source you want the function generator to use. The function generator derives the frequencies and sample rates that it uses to generate waveforms from the source you specify. Options include Internal, External, and RTSI Clock. For example, when you set Reference clock source to External, the function generator uses the signal it receives at its external clock terminal as its reference clock.

</td></tr>
<tr><td>Execution Control</td><td>Contains the following execution control options:

Start this step after—Makes the step wait until another step has started before executing. You can make the step wait on any other hardware step in the project by selecting the step to wait on from the pull-down menu.

You can use this option to force an acquisition device to start after a generation device starts. You also can use this option to ensure that a device generating a trigger signal starts after the device receiving the signal, which avoids sending the signal before the receiver is ready.
Step to wait for—Lists the possible steps for which this step can wait.
Pre-execution delay (ms)—Specifies the amount of time to wait before the step executes. If you configure the step to start after another

</td></tr>
</table>

	step, the delay represents the amount of time to wait after the specified step starts. • **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes.

IVI Power Supply

Generates a voltage level using an instrument in the DC Power Supply IVI class.

Default values are specific to the hardware and driver specified. The default settings may not be applicable to the measurement you are trying to perform. Click the **Initialize** button, located on the step's toolbar, at any time to reset the step to the default settings.

To communicate with an instrument, you need to install the instrument-specific driver and create a session name for the instrument.

Parameter	Description
Query device for measurement	Queries the power supply for the actual current and voltage the device is generating. **Note** When you place a checkmark in the **Query device for measurement** checkbox, the performance of the device decreases.
Voltage Display	Displays the voltage (V) of the power supply.
Current Display	Displays the current (A) of the power supply.
Output	Contains the following output options: • **Export over-voltage tripped**—Exports whether the over-voltage was tripped as a Boolean value in the Project View. • **Export over-current tripped**—Exports whether the over-current was tripped as a Boolean value in the Project View.
Configuration	Contains the following power supply configuration options: • **Device**—Contains the following device options: – **IVI session name**—Specifies the session name to use for this step. This step retrieves possible session names from National Instruments Measurement & Automation Explorer (MAX). You also

can create a new session or edit/delete an existing session.
  - **Resource descriptor**—Specifies the interface and the address of the hardware to associate with the step.
  - **Instrument driver**—Displays the name of the driver in use.
- **Channel Configuration**—Contains the following channel configuration options:
  - **Channels**—Specifies the physical channels from which to generate data.
  - **Enable channel**—Specifies whether to enable data acquisition on the selected channel.
  - **Voltage level (V)**—Specifies the DC voltage you want the power supply to attempt to generate.
  - **OVP enabled**—Specifies whether you want to use an over-voltage protection limit. Place a checkmark in this checkbox to enable the **OVP limit (V)** field.
  - **OVP limit (V)**—Specifies the over-voltage protection limit you want to use.
  - **Specify output range**—Select this option to define an output range for the output current or voltage.
  - **Range type**—Specifies the type of range to configure.
  - **Range (A)**—Specifies the range in amperes.
  - **Range (V)**—Specifies the value of the input range the oscilloscope uses for the channel. For example, to acquire a sine wave that spans -5 to 5 volts, enter 10 as the value of this parameter.
  - **Current limit behavior**—Specifies the

	behavior you want the power supply to exhibit when the output current is greater than or equal to the value of **Current limit (A)**. Options include Regulate and Trip. – **Current limit (A)**—Specifies the current limit you want to use.
Trigger	Contains the following trigger options: **Trigger generation**—Specifies whether any channels wait for triggers. **Trigger generation** is disabled by default. If you do not enable **Trigger generation**, the power supply generates the current and **Voltage level (V)** when you click the **Run** button. Place a checkmark in this checkbox to enable the **Channel Triggering** options and configure triggers. **Channel Triggering**—Contains the following channel trigger options: – **Channels**—Specifies the physical channels that have a trigger associated with them. Each channel can be triggered independently. – **Type**—Specifies the trigger source to which you want the power supply to respond. The default is Immediate. Contains the following options: **Immediate**—Does not wait for a trigger of any kind. **External**—Waits for a trigger on the external trigger input. **Software**—Waits until the software trigger button is pressed from the toolbar, specified by the software trigger source. **Digital**—Waits for a trigger on a digital input specified by the trigger source.

- **Triggered level (V)**—Specifies the DC voltage level you want the power supply to attempt to generate after it receives a trigger.
- **Triggered current limit (A)**—Specifies the current limit you want the power supply to use after it receives a trigger.
- **Software trigger source**—Specifies the software trigger source to which you want the instrument to respond. Options include Trigger A, Trigger B, and Trigger C. The default is Trigger A. **Software trigger source** is available only when you select **Software** in **Type**.
- **Source**—Specifies the trigger source you want to use. **Source** is available only when you select **Digital** in **Type**. Contains the following options:
    - **PXI TRIG0 or VXI TTL0**—Waits until it receives a trigger on the PXI TRIG0 line (for PXI instruments) or the VXI TTL0 line (for VXI instruments).
    - **PXI TRIG1 or VXI TTL1**—Waits until it receives a trigger on the PXI TRIG1 line (for PXI instruments) or the VXI TTL1 line (for VXI instruments).
    - **PXI TRIG2 or VXI TTL2**—Waits until it receives a trigger on the PXI TRIG2 line (for PXI instruments) or the VXI TTL2 line (for VXI instruments).
    - **PXI TRIG3 or VXI TTL3**—Waits until it receives a trigger on the PXI TRIG3 line (for PXI instruments) or the VXI TTL3 line

		(for VXI instruments). • **PXI TRIG4 or VXI TTL4**—Waits until it receives a trigger on the PXI TRIG4 line (for PXI instruments) or the VXI TTL4 line (for VXI instruments). • **PXI TRIG5 or VXI TTL5**—Waits until it receives a trigger on the PXI TRIG5 line (for PXI instruments) or the VXI TTL5 line (for VXI instruments). • **PXI TRIG6 or VXI TTL6**—Waits until it receives a trigger on the PXI TRIG6 line (for PXI instruments) or the VXI TTL6 line (for VXI instruments). • **PXI TRIG7 or VXI TTL7**—Waits until it receives a trigger on the PXI TRIG7 line (for PXI instruments) or the VXI TTL7 line (for VXI instruments). • **ECL0**—Waits until it receives a trigger on the VXI ECL0 line. • **ECL1**—Waits until it receives a trigger on the VXI ECL1 line. • **PXI Star**—Waits until it receives a trigger on the PXI STAR trigger bus. • **RTSI 0**—Waits until it receives a trigger on RTSI line 0. • **RTSI 1**—Waits until it receives a trigger on RTSI line 1. • **RTSI 2**—Waits until it receives a trigger on RTSI line 2. • **RTSI 3**—Waits until it receives a trigger on RTSI line 3. • **RTSI 4**—Waits until it receives a

	trigger on RTSI line 4. • **RTSI 5**—Waits until it receives a trigger on RTSI line 5. • **RTSI 6**—Waits until it receives a trigger on RTSI line 6.
Execution Control	Contains the following execution control options: • **Start this step after**—Makes the step wait until another step has started before executing. You can make the step wait on any other hardware step in the project by selecting the step to wait on from the pull-down menu. You can use this option to force an acquisition device to start after a generation device starts. You also can use this option to ensure that a device generating a trigger signal starts after the device receiving the signal, which avoids sending the signal before the receiver is ready. • **Step to wait for**—Lists the possible steps for which this step can wait. • **Pre-execution delay (ms)**—Specifies the amount of time to wait before the step executes. If you configure the step to start after another step, the delay represents the amount of time to wait after the specified step starts. • **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes.

Create Signals

Use the Create Signals steps to create different types of standard periodic signals, noise, multi-tone, or DC signals.

 To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Create Analog Signal

Creates an analog signal. You can create various periodic waveform signals as well as noise, multi-tone, or DC signals. You also can use a formula to define a signal. Use Create Analog Signal to create arbitrary signals. For example, you can use Create Analog Signal to create a signal to use as a stimulus for a hardware device. Create Analog Signal can run in continuous signal mode or repeated signal mode, depending on whether you place a checkmark in the **Repeated signal** checkbox. The default is continuous signal mode.

Details

Parameter	Description
Output Signal	Displays the signal the step creates.
Configuration	Contains the following options: • **Signal Calculation Setup**—Contains options you can use to configure how the step calculates the signal. The options that appear in this section depend on the **Signal type** you specify. Contains the following options: – **Signal type**—Specifies the type of signal Create Analog Signal creates. You can select from the following options: • **Sine Wave**—(Default) Creates a sine wave with a default amplitude and frequency of 1. • **Triangle Wave**—Creates a triangle wave with a default amplitude and frequency of 1. • **Square Wave**—Creates a square wave with a default amplitude and frequency of 1. • **Sawtooth Wave**—Creates a sawtooth wave with a default amplitude and frequency of 1. • **DC Signal**—Creates a DC signal with a default offset of 0V.

- **Noise Signal**—Creates a noise signal with a default level of 1. Use the **Noise type** field to specify Gaussian, rectangular, or triangular amplitude distribution.
- **Multi-tone**—Creates a multi-tone signal with a default start frequency and amplitude of 1 and a default stop frequency of 2. Multi-tone signals allow the fast and efficient stimulus of a system across an arbitrary band of frequencies, and you can use them to determine the frequency response of a device.
- **Formula**—Creates a signal according to the formula you enter in the **Formula** field.

– **Frequency (Hz)**—[Signal type: Sine Wave, Triangle Wave, Square Wave, Sawtooth Wave, Formula] Specifies the frequency of a sine, triangle, square, or sawtooth wave in hertz or the value of f if you select the **Formula** signal type. The default is 1.0 kHz.

– **Amplitude (V)**—[Signal type: Sine Wave, Triangle Wave, Square Wave, Sawtooth Wave, Formula] Specifies the amplitude of a sine, triangle, square, or sawtooth wave or the value of a if you select the **Formula** signal type. The default is 1.0 V.

– **Phase (deg.)**—[Signal type: Sine Wave, Triangle Wave, Square Wave, Sawtooth Wave] Specifies the initial phase of a sine, triangle, square, or sawtooth wave in degrees. The default is 0 degrees.

– **Offset (V)**—Specifies the DC offset of the signal. The default is 0 V.

- **Repeated signal**—Specifies if the created signal is repeated or continuous. If you place a checkmark in this checkbox, Create Analog Signal calculates the signal only during the first iteration of the step after you click the **Run** button or the **Reset Signal** button and each time you change a configuration parameter. The signal then repeats with the same time stamp and start phase.
- **N periods**—[Signal type: Sine Wave, Triangle Wave, Square Wave, Sawtooth Wave] Forces the number of periods in the signal to be an integer. If you select this option and change the values of **Sample rate (S/s)** or **Block size (samples)**, Create Analog Signal coerces the value of **Frequency (Hz)** so that the number of periods remains an integer.
- **Duty cycle (%)**—[Signal type: Square Wave] Specifies the percentage of each period a square wave remains high.
- **Noise type**—[Signal type: Noise Signal] Specifies the type of noise the probability density function represents. Create Analog Signal defines the **Noise type** by the distribution of frequencies that appear on a histogram of the signal.
    - **White (Gaussian)**—(Default) Creates a noise signal with a Gaussian distribution of frequencies.
    - **White (Rectangular)**—Creates a noise signal with a rectangular distribution of frequencies.

- **White (Triangular)**—Creates a noise signal with a triangular distribution of frequencies.

- **Level (Vrms)**—[Signal type: Noise Signal] Specifies the noise level. The default is 1 V_rms. This option is available only when you select **White (Gaussian)** as the **Noise type**.
- **Start freq. (Hz)**—[Signal type: Multi-tone] Specifies the start frequency of the multi-tone signal. This step coerces the start frequency to be a multiple of the frequency resolution defined by the ratio of the **Sample rate (S/s)** divided by the **Block size (samples)**.
- **Stop freq. (Hz)**—[Signal type: Multi-tone] Specifies the stop frequency of the multi-tone signal. This step coerces the stop frequency to equal **Start freq. (Hz)** + *n* * **Step freq. (Hz)**, where *n* is an integer number.
- **Step freq. (Hz)**—[Signal type: Multi-tone] Specifies the step frequency of the multi-tone signal. This step coerces the step frequency to be a multiple of the frequency resolution defined by the ratio of the **Sample rate (S/s)** divided by the **Block size (samples)**.
- **Formula**—[Signal type: Formula] Specifies the formula string that defines the signal. The default is a*sin(w*t). You can use the following defined variable names:
  - f—Frequency equal to the **Frequency (Hz)** input.
  - a—Amplitude equal to the **Amplitude (V)** input.

	- w—2*pi*f. - n—Current number of samples generated. - t—Number of elapsed seconds. - fs—Sampling frequency equal to the **Sample rate (S/s)**. **Sampling Conditions**—Contains the following options: – **Sample rate (S/s)**—Specifies the sampling rate of the signal in samples per second. The default is 100 kS/s. – **Block size (samples)**—Specifies the number of samples in the signal. The default is 1000 samples. **Optional Outputs**—Contains the following option: – **Export coerced values**—Exports coerced frequency values as output scalar values. Create Analog Signal can coerce frequency values when you select a periodic signal type and you place a checkmark in the **N periods** checkbox, or when you select a multi-tone signal type.
Execution Control	Contains the following option: **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes.

Create Analog Signal Details

In continuous signal mode, the signal that Create Analog Signal creates at each iteration is contiguous to the previous iteration. The result is a signal with a continuously increasing time stamp and phase continuity. You can use the continuous signal mode to continuously generate a signal with arbitrary frequency or a non-repetitive noise signal with an analog output device if the device supports updating the output buffer while running.

In repeated signal mode, this step calculates the signal only during the first iteration of the step after you click the **Run** button or, in LabVIEW SignalExpress, the **Reset Signal** button and each time you change a configuration parameter. The signal then repeats with the same time stamp and start phase. National Instruments recommends that you use repeated signal mode if you generate the signal with an analog output device that does not support updating the output buffer while running, such as the devices that NI-FGEN Arbitrary Waveform supports.

Signal Frequency Coercion

If you select a standard periodic signal type, you can place a checkmark in the **N periods** checkbox to coerce the signal frequency you specify so that Create Analog Signal creates a signal with an integer number of periods. The periods repeat without phase discontinuities. If you place a checkmark in the **N periods** checkbox, the actual coerced values overwrite the input value you type. When you select a multi-tone signal type, Create Analog Signal coerces the **Start freq. (Hz)**, **Stop freq. (Hz)**, and **Step freq. (Hz)** values to create a repeatable signal. Place a checkmark in the **Export coerced values** checkbox to export coerced frequency values as an output of Create Analog Signal.

Create Digital Signal

Creates different types of digital signals. Depending on the option you select in the **Signal type** pull-down menu, this step can create a ramp, marching values, single value, random, or toggle pattern.

Parameter	Description
Output Signal	Displays the signal the step creates.
Configuration	Contains the following options: • **Signal type**—Specifies the type of digital waveform to create. You can select from the following options: – **Ramp**—Creates a digital waveform that contains a binary count-up pattern that starts at zero and counts up by one until it reaches 2^{n-1}, where *n* = **Number of signals**. – **Marching Values**—Creates a digital waveform in which a binary value placed on the first signal of the first sample is logically shifted to the next signal on each subsequent sample of the waveform. The **Hold value** field specifies the initial value, and the **Marching value** field specifies how the value shifts for subsequent samples. – **Single Value**—Creates a digital waveform in which all bits are set to 0, 1, Z, L, H, X, T, or V, depending on the **Value** you specify. – **Random**—Creates a digital waveform that contains a random digital pattern of 0s and 1s. The random pattern generated assumes no mathematically determinable sequence of values. – **Toggle**—Creates a digital waveform in which the even numbered samples

contain binary values you define in the **Toggle value 1** field and the odd numbered samples contain binary values you define in the **Toggle value 2** field.

- **Hold value**—Specifies the binary value of the generated digital waveform. This option is only available when you select the **Marching Values** option from the **Signal type** pull-down menu.
- **Marching value**—Specifies the binary value that marches across the signals of the generated digital waveform. This option is only available when you select the **Marching Values** option from the **Signal type** pull-down menu.
- **Value**—Specifies the digital bit state of the generated digital waveform. This option is available only when you select the **Signal Value** option from the **Signal type** pull-down menu.
- **Toggle value 1**—Specifies the first digital bit state of the generated digital waveform. This option is available when you select the **Toggle** option from the **Signal type** pull-down menu.
- **Toggle value 2**—Specifies the second digital bit state of the generated digital waveform. This option is available when you select the **Toggle** option from the **Signal type** pull-down menu.
- **Create one output per signal**—Specifies whether to create an output group that contains a separate signal for each line in the digital waveform.
- **Number of signals**—Specifies the number of signals to include in the generated digital waveform.
- **Block size (samples)**—Specifies the number of samples in the signal. The default is 1000 samples.
- **Sample rate (S/s)**—Specifies the sampling rate of the signal in samples per second. The default

	is 100 kS/s.
Execution Control	Contains the following option: • **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes.
Signal Names	Contains the following options: • **Signal names table**—Specifies custom names for the digital signals this step creates. • **Reset to default**—Resets any modified digital signal names to its default name.

Load/Save Signals

Use the Load/Save Signals steps to import or export data from ASCII and LVM files; and import data from SPICE files.

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Load from ASCII (Frequency Domain)

Imports data from an ASCII file.

Parameter	Description
Imported Signal	Displays the signal imported from an ASCII file.

Parse File	Contains the following options:
	• **Import file path**—Specifies the name and location of the file you want to import. You can specify an absolute or relative path to the file. If you specify an absolute path, this step saves the path with the project. If you specify a relative path and you do not save the project, this step assumes the path is relative to the **My Documents** folder. If you specify a relative path and you save the project, the path is relative to the location where you save the project.
	• **File preview**—Displays a preview of the contents of the file to help you determine how to parse the parameters. By default, **File preview** displays the first 50 rows from the file. If you increase the value in **Start row**, **File preview** displays 50 rows beginning with the row you specify in **Start row**. You can resize the column header to show more or less of the column.
	• **File Parsing Settings**—Contains the following options:
	  – **Delimiter**—Specifies the delimiter to use to separate data. The default is Tab. This option appears only if you set **Export file type** to **Generic ASCII (.txt)**.
	  – **Custom Delimiter**—Specifies a file delimiter other than a tab or a comma. This option appears only if you set **Export file type** to **Generic ASCII (.txt)**.
	  – **Start row**—Specifies the row from which to begin displaying data. The default is 1.

<table>
<tr>
<td></td>
<td>

End row—Specifies the last row to display data. The default is –1, which specifies to display all data.
Signal names precede data row—Specifies that the first row of the file contains the signal names.
Decimal point—Specifies which character to use as the decimal point. The default is . (dot).
Domain—Specifies the data type of the output signal.

</td>
</tr>
<tr>
<td>Import Signals</td>
<td>Contains the following options:

Signals—Lists the signals in the file and specifies which signal the Imported Signal graph displays.
Import—Specifies whether to import the selected signal from the file you specify in Import file path. If you place a checkmark in one of the Import checkboxes, LabVIEW SignalExpress exports that signal to the Project View. You can then send that signal to another step or plot it on the Data View.
Resampling Setup—Contains the following options:
Note When you select an input channel as your X value and specify your own df, LabVIEW SignalExpress resamples the waveform according to the new df and based on the selected X values. This may change the number of data points in the output waveform.

Input X values—Specifies the X data to use for resampling. Options include None (default) or Point Index.
Interpolation mode—Specifies the interpolation method. Options include Coerce, Linear, or Spline. The default is Coerce.
User specified df—Specifies the interval

</td>
</tr>
</table>

		size that represents the sampling step size to use to obtain data. – **Use same df**—Specifies whether to use the same df for all **Signals**. When you place a checkmark in this checkbox, Load from ASCII applies the df of the currently selected signal to all **Signals**. Select this option to group the imported **Signals** into one output. – **Signal Type**—Defines the type of frequency signal. Options include: Magnitude - linear, Magnitude - dB, Phase - degrees, or Phase - radians.

Load from ASCII (Time Domain)

Imports data from an ASCII file.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Imported Signal</td><td>Displays the signal imported from an ASCII file.</td></tr>
<tr><td>Parse File</td><td>Contains the following options:

Import file path—Specifies the name and location of the file you want to import. You can specify an absolute or relative path to the file. If you specify an absolute path, this step saves the path with the project. If you specify a relative path and you do not save the project, this step assumes the path is relative to the My Documents folder. If you specify a relative path and you save the project, the path is relative to the location where you save the project.
File preview—Displays a preview of the contents of the file to help you determine how to parse the parameters. By default, File preview displays the first 50 rows from the file. If you increase the value in Start row, File preview displays 50 rows beginning with the row you specify in Start row. You can resize the column header to show more or less of the column.
File Parsing Settings—Contains the following options:

Delimiter—Specifies the delimiter to use to separate data. The default is Tab. This option appears only if you set Export file type to Generic ASCII (.txt).
Custom Delimiter—Specifies a file delimiter other than a tab or a comma. This option appears only if you set Export file type to Generic ASCII (.txt).
Start row—Specifies the row from which to begin displaying data. The default is 1.

</td></tr>
</table>

<table>
<tr>
<td></td>
<td>

End row—Specifies the last row to display data. The default is –1, which specifies to display all data.
Signal names precede data row—Specifies that the first row of the file contains the signal names.
Decimal point—Specifies which character to use as the decimal point. The default is . (dot).
Domain—Specifies the data type of the output signal.

</td>
</tr>
<tr>
<td>Import Signals</td>
<td>Contains the following options:

Signals—Lists the signals in the file and specifies which signal the Imported Signal graph displays.
Import—Specifies whether to import the selected signal from the file you specify in Import file path. If you place a checkmark in one of the Import checkboxes, LabVIEW SignalExpress exports that signal to the Project View. You can then send that signal to another step or plot it on the Data View.
Resampling Setup—Contains the following options:

Note When you select an input channel as your X value and specify your own dt, LabVIEW SignalExpress resamples the waveform according to the new dt and based on the selected X values. This may change the number of data points in the output waveform.

Input X values—Specifies the X data to use for resampling. Options include None (default) or Point Index.
Interpolation mode—Specifies the interpolation method. Options include Coerce, Linear, or Spline. The default is Coerce.
User specified dt—Specifies the interval

</td>
</tr>
</table>

		size that represents the sampling step size to use to obtain data. – **Use same dt**—Indicates whether to use the same dt for all **Signals**. When you place a checkmark in this checkbox, Load from ASCII applies the dt of the currently selected signal to all **Signals**. Select this option to group the imported **Signals** into one output. • **Y axis unit**—Specifies the unit type for the input signal. Selecting **Unitless** specifies to not associate any unit with the input signal. Selecting **Custom** specifies that the input signal contains an associated unit within the data. • **X axis unit**—Specifies the unit type for the input signal. Selecting **Unitless** specifies to not associate any unit with the input signal. Selecting **Custom** specifies that the input signal contains an associated unit within the data.

Load from LVM (Frequency Domain)

Imports data from a text-based measurement file (.lvm).

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Imported Signal</td><td>Displays the signal imported from the .lvm file.</td></tr>
<tr><td>File and Signal Selection</td><td>Contains the following options:

Import file path—Specifies the name and location of the file you want to import. You can specify an absolute or relative path to the file. If you specify an absolute path, this step saves the path with the project. If you specify a relative path and you do not save the project, this step assumes the path is relative to the My Documents folder. If you specify a relative path and you save the project, the path is relative to the location where you save the project.
Signals—Lists the signals in the file and specifies which signal the Imported Signal graph displays.
Import—Specifies whether to import the selected signal from the file you specify in Import file path. If you place a checkmark in one of the Import checkboxes, LabVIEW SignalExpress exports that signal to the Project View. You can then send that signal to another step or plot it on the Data View.
File Information—Contains the following options:

ID—Contains the following options:

Project—Displays the name of the project associated with this data set.
User—Displays the identity of the user who saved this data.
Save date—Indicates the date when the data was saved.
Save time—Indicates the time when the data was saved.

Notes—Specifies miscellaneous information related to the saved data.

</td></tr>
</table>

    - **UUT**—Contains the following options:
        - **UUT name or description**—Specifies the name and/or description of the Unit Under Test (UUT).
        - **Serial number**—Specifies the serial number of the UUT.
        - **Model number**—Specifies the model number of the UUT.
    - **Test**—Specifies testing information to add to the header. Contains the following options:
        - **Test name or description**—Specifies the name and/or description of the test.
        - **Series**—Specifies the test series of this data.
        - **Numbers**—Specifies the numbers in the test series to which this data corresponds.
    - **Waveform**—Displays information about the waveform.
        - **Waveform name**—Specifies the name of the channel you select.
        - **Waveform notes**—Specifies miscellaneous information associated with the channel you select.
- **Domain**—Specifies the data type of the output signal. The default is Frequency Waveform.
- **Signal Type**—Defines the type of frequency signal. Options include: Magnitude - linear, Magnitude - dB, Phase - degrees, or Phase - radians.

Load from LVM (Time Domain)

Imports data from a text-based measurement file (.lvm).

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Imported Signal</td><td>Displays the signal imported from the .lvm file.</td></tr>
<tr><td>File and Signal Selection</td><td>Contains the following options:

Import file path—Specifies the name and location of the file you want to import. You can specify an absolute or relative path to the file. If you specify an absolute path, this step saves the path with the project. If you specify a relative path and you do not save the project, this step assumes the path is relative to the My Documents folder. If you specify a relative path and you save the project, the path is relative to the location where you save the project.
Signals—Lists the signals in the file and specifies which signal the Imported Signal graph displays.
Import—Specifies whether to import the selected signal from the file you specify in Import file path. If you place a checkmark in one of the Import checkboxes, LabVIEW SignalExpress exports that signal to the Project View. You can then send that signal to another step or plot it on the Data View.
File Information—Contains the following options:

ID—Contains the following options:

Project—Displays the name of the project associated with this data set.
User—Displays the identity of the user who saved this data.
Save date—Indicates the date when the data was saved.
Save time—Indicates the time when the data was saved.

Notes—Specifies miscellaneous information related to the saved data.

</td></tr>
</table>

- **UUT**—Contains the following options:
  - **UUT name or description**—Specifies the name and/or description of the Unit Under Test (UUT).
  - **Serial number**—Specifies the serial number of the UUT.
  - **Model number**—Specifies the model number of the UUT.
- **Test**—Specifies testing information to add to the header. Contains the following options:
  - **Test name or description**—Specifies the name and/or description of the test.
  - **Series**—Specifies the test series of this data.
  - **Numbers**—Specifies the numbers in the test series to which this data corresponds.
- **Waveform**—Displays information about the waveform.
  - **Waveform name**—Specifies the name of the channel you select.
  - **Waveform notes**—Specifies miscellaneous information associated with the channel you select.

- **Domain**—Specifies the data type of the output signal.
- **Y axis unit**—Specifies the unit type for the input signal. Selecting **Unitless** specifies to not associate any unit with the input signal. Selecting **Custom** specifies that the input signal contains an associated unit within the data.
- **X axis unit**—Specifies the unit type for the input signal. Selecting **Unitless** specifies to not associate

		any unit with the input signal. Selecting **Custom** specifies that the input signal contains an associated unit within the data.

Load from SPICE (Frequency Domain XY)

Imports data from a SPICE, PSpice, or Multisim file.

SPICE is a general-purpose circuit simulation program for nonlinear DC, nonlinear transient, and linear AC analyses. PSpice is SPICE for Windows and is part of the OrCAD® product line by Cadence Design Systems Inc. Multisim is an integrated desktop design entry and simulation system for design engineers created by Electronics Workbench.

Parameter	Description
Imported Signal	Displays the signal you imported from a SPICE, PSpice, or Multisim file.
File and Signal Selection	Contains the following options: • **Simulation file type**—Specifies the type of file to import. Options include SPICE, PSpice, and Multisim. The default is SPICE. • **Import file path**—Specifies the name and location of the file you want to import. You can specify an absolute or relative path to the file. If you specify an absolute path, this step saves the path with the project. If you specify a relative path and you do not save the project, this step assumes the path is relative to the **My Documents** folder. If you specify a relative path and you save the project, the path is relative to the location where you save the project. • **Signals in file**—Lists the signals in the file and specifies which signal the **Imported Signal** graph displays. • **Import**—Specifies whether to import the selected signal from the file you specify in **Import file path**. If you place a checkmark in one of the **Import** checkboxes, LabVIEW SignalExpress exports that signal to the Project View. You can then send that signal to another step or plot it on the Data View. • **Signal Type**—Defines the type of frequency signal. Options include: Magnitude - linear, Magnitude - dB,

		Phase - degrees, or Phase - radians. • **Domain**—Specifies the data type of the output signal. The default is XY Pairs-Frequency. • **Magnitude**—Specifies the data type of the output signal as **Linear** or **dB**. This option is available only when the file you import contains complex signals that include real and imaginary components. • **Phase**—Specifies the data type of the output signal as **degrees** or **radians**. This option is available only when the file you import contains complex signals that include real and imaginary components.

Load from SPICE (Frequency Domain)

Imports data from a SPICE, PSpice, or Multisim file.

SPICE is a general-purpose circuit simulation program for nonlinear DC, nonlinear transient, and linear AC analyses. PSpice is SPICE for Windows and is part of the OrCAD® product line by Cadence Design Systems Inc. Multisim is an integrated desktop design entry and simulation system for design engineers created by Electronics Workbench.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Imported Signal</td><td>Displays the signal you imported from a SPICE, PSpice, or Multisim file.</td></tr>
<tr><td>File and Signal Selection</td><td>Contains the following options:

Simulation file type—Specifies the type of file to import. Options include SPICE, PSpice, and Multisim. The default is SPICE.
Import file path—Specifies the name and location of the file you want to import. You can specify an absolute or relative path to the file. If you specify an absolute path, this step saves the path with the project. If you specify a relative path and you do not save the project, this step assumes the path is relative to the My Documents folder. If you specify a relative path and you save the project, the path is relative to the location where you save the project.
Signals in file—Lists the signals in the file and specifies which signal the Imported Signal graph displays.
Import—Specifies whether to import the selected signal from the file you specify in Import file path. If you place a checkmark in one of the Import checkboxes, LabVIEW SignalExpress exports that signal to the Project View. You can then send that signal to another step or plot it on the Data View.
Resampling Setup—Contains the following options:

Interpolation mode—Specifies the

</td></tr>
</table>

		interpolation method. Options include Coerce, Linear, or Spline. The default is Coerce. – **df**—Specifies the df. The default is 1. – **Signal Type**—Defines the type of frequency signal. Options include: Magnitude - linear, Magnitude - dB, Phase - degrees, or Phase - radians. – **Magnitude**—Specifies the data type of the output signal as **Linear** or **dB**. This option is available only when the file you import contains complex signals that include real and imaginary components. – **Phase**—Specifies the data type of the output signal as **degrees** or **radians**. This option is available only when the file you import contains complex signals that include real and imaginary components. • **Domain**—Specifies the data type of the output signal. The default is Frequency Waveform.

Load from SPICE (Time Domain XY)

Imports data from a SPICE, PSpice, or Multisim file.

SPICE is a general-purpose circuit simulation program for nonlinear DC, nonlinear transient, and linear AC analyses. PSpice is SPICE for Windows and is part of the OrCAD® product line by Cadence Design Systems Inc. Multisim is an integrated desktop design entry and simulation system for design engineers created by Electronics Workbench.

Parameter	Description
Imported Signal	Displays the signal you imported from a SPICE, PSpice, or Multisim file.
File and Signal Selection	Contains the following options: • **Simulation file type**—Specifies the type of file to import. Options include SPICE, PSpice, and Multisim. The default is SPICE. • **Import file path**—Specifies the name and location of the file you want to import. You can specify an absolute or relative path to the file. If you specify an absolute path, this step saves the path with the project. If you specify a relative path and you do not save the project, this step assumes the path is relative to the **My Documents** folder. If you specify a relative path and you save the project, the path is relative to the location where you save the project. • **Signals in file**—Lists the signals in the file and specifies which signal the **Imported Signal** graph displays. • **Import**—Specifies whether to import the selected signal from the file you specify in **Import file path**. If you place a checkmark in one of the **Import** checkboxes, LabVIEW SignalExpress exports that signal to the Project View. You can then send that signal to another step or plot it on the Data View. • **Domain**—Specifies the data type of the output signal. The default is XY Pairs-Time.

		• **Y axis unit**—Specifies the unit type for the input signal. Selecting **Unitless** specifies to not associate any unit with the input signal. Selecting **Custom** specifies that the input signal contains an associated unit within the data. • **X axis unit**—Specifies the unit type for the input signal. Selecting **Unitless** specifies to not associate any unit with the input signal. Selecting **Custom** specifies that the input signal contains an associated unit within the data.

Load from SPICE (Time Domain)

Imports data from a SPICE, PSpice, or Multisim file.

SPICE is a general-purpose circuit simulation program for nonlinear DC, nonlinear transient, and linear AC analyses. PSpice is SPICE for Windows and is part of the OrCAD® product line by Cadence Design Systems Inc. Multisim is an integrated desktop design entry and simulation system for design engineers created by Electronics Workbench.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Imported Signal</td><td>Displays the signal you imported from a SPICE, PSpice, or Multisim file.</td></tr>
<tr><td>File and Signal Selection</td><td>Contains the following options:

Simulation file type—Specifies the type of file to import. Options include SPICE, PSpice, and Multisim. The default is SPICE.
Import file path—Specifies the name and location of the file you want to import. You can specify an absolute or relative path to the file. If you specify an absolute path, this step saves the path with the project. If you specify a relative path and you do not save the project, this step assumes the path is relative to the My Documents folder. If you specify a relative path and you save the project, the path is relative to the location where you save the project.
Signals in file—Lists the signals in the file and specifies which signal the Imported Signal graph displays.
Import—Specifies whether to import the selected signal from the file you specify in Import file path. If you place a checkmark in one of the Import checkboxes, LabVIEW SignalExpress exports that signal to the Project View. You can then send that signal to another step or plot it on the Data View.
Resampling Setup—Contains the following options:

Interpolation mode—Specifies the

</td></tr>
</table>

	interpolation method. Options include Coerce, Linear, or Spline. The default is Coerce. – **dt**—Specifies the dt. The default is 1. – **Y-axis**—Specifies to display the y-axis as **linear** or **dB**. • **Domain**—Specifies the data type of the output signal. • **Y axis unit**—Specifies the unit type for the input signal. Selecting **Unitless** specifies to not associate any unit with the input signal. Selecting **Custom** specifies that the input signal contains an associated unit within the data. • **X axis unit**—Specifies the unit type for the input signal. Selecting **Unitless** specifies to not associate any unit with the input signal. Selecting **Custom** specifies that the input signal contains an associated unit within the data.

Save to ASCII/LVM

Saves a signal to an ASCII file or a text-based measurement file (.lvm). Because the .lvm file format is designed to never overwrite the initial header information, Save to ASCII/LVM only writes the header information to the .lvm file the first time you run. If you run again, Save to ASCII/LVM does not update the header information, only the subhead information.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Input Signals</td><td>Displays the input signals.</td></tr>
<tr><td>Signals</td><td>Contains the following options:
Add Input—Adds the signal you want to write to file.
Remove—Removes the highlighted input.
Inputs—Lists the signals to write to the ASCII file or the text-based measurement file (.lvm).
Input Data—Specifies the signal you want to save to the file.</td></tr>
<tr><td>File Settings</td><td>Contains the following options:
Export file path—Specifies the location where you want to save the file. You can specify an absolute or relative path to the file. If you specify an absolute path, this step saves the path with the project. If you specify a relative path and you do not save the project, this step assumes the path is relative to the My Documents folder. If you specify a relative path and you save the project, the path is relative to the location where you save the project.
If file already exists—Specifies how LabVIEW SignalExpress saves data to an existing file. Contains the following options:
Overwrite—Replaces data in the existing file.
Overwrite once, then append to file—Overwrites the file once and then appends to information to the end of the existing file.</td></tr>
</table>

- **Overwrite and backup previous**—Performs a backup of the file and replaces data in the existing file.
- **Append to file**—Appends the data to the existing file.
- **Next available file name**—Appends the next sequential number to the filename. For example, if test.lvm exists, LabVIEW SignalExpress saves the file as test1.lvm.

- **Export file type**—Specifies in which file format to save the file. Options include text-based measurement file (.lvm) or Generic ASCII. The default is text-based measurement file (.lvm). If you select Generic ASCII, you also can save as .csv or .txt. A .lvm file format contains header information and signal data. An ASCII file format contains only signal data. Both file formats are ASCII. You can use the Load from LVM or Load from ASCII steps to load these files into LabVIEW SignalExpress.
- **LVM File Annotations**—Contains the following options:
  - **ID**—Contains the following options:
    - **Project**—Displays the name of the project associated with this data set.
    - **User**—Displays the identity of the user who saved this data.
  - **Notes**—Specifies miscellaneous information associated with the channel you select.
  - **UUT**—Contains the following options:
    - **UUT name or description**—Specifies the name and/or description of the Unit Under Test (UUT).
    - **Serial number**—Specifies the serial number of the UUT.
    - **Model number**—Specifies the model number of the UUT.

- **Test**—Specifies testing information to add to the header. Contains the following options:
  - **Test name or description**—Specifies the name and/or description of the test.
  - **Series**—Specifies the test series of this data.
  - **Numbers**—Specifies the numbers in the test series to which this data corresponds.
- **Waveform**—Displays information about the waveform.
  - **Waveform name**—Specifies the name of the channel you select.
  - **Waveform notes**—Specifies miscellaneous information associated with the channel you select.

- **Delimiter**—Specifies the delimiter to use to separate data. The default is Tab. This option appears only if you set **Export file type** to **Generic ASCII (.txt)**.
- **Custom Delimiter**—Specifies a file delimiter other than a tab or a comma. This option appears only if you set **Export file type** to **Generic ASCII (.txt)**.
- **Include Signal Names**—Includes the names of the signals in the ASCII file. This option appears only if you set **Export file type** to **Generic ASCII (.txt)**.
- **X Value Columns**—Contains the following options:
  - **One column per channel**—Creates a separate column for time data each channel generates. This option includes a column of values from the x-axis for every column of values from the y-axis.
  - **One column only**—Creates only one column for the time data the channels generate. This option includes only one column of values from the x-axis.

- **Empty time column**—Creates an empty column for the time data each channel generates. This option does not include the data from the x-axis.

This option appears only if you set **Export file type** to **Generic ASCII (.txt)**.

- **Time Axis Preference**—Contains the following options:
  - **Absolute Time**—Displays the time elapsed since 12:00 a.m., Friday, January 1, 1904, Universal Time.
  - **Relative Time**—Displays the time in milliseconds starting from 0.

  This option appears only if you set **Export file type** to **Generic ASCII (.txt)**.

Processing

Use the Processing steps to filter, scale, resample, and average signals; apply windowing and perform arithmetic operations; and, interactively align two signals.

 To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Filter

Filters a time signal using an infinite impulse response (IIR) or finite impulse response (FIR) filter. Use this step to remove or attenuate unwanted frequencies from a signal using various standard filter types and topologies.

In LabVIEW SignalExpress, the Filter step filters the input signal continuously. The step resets the signal to its original value the first time the step runs, if LabVIEW SignalExpress detects a discontinuity in the input signal, or if you press the **Reset Filter** button.

In LabVIEW, the Filter Express VI filters the input signal continuously. The Express VI resets the signal to its original value the first time the Express VI runs, if LabVIEW detects a discontinuity in the input signal, or if the **reset** input receives a TRUE value.

Details

Parameter	Description
Input Signals	Displays the input signal to filter.
Autoscale amplitude	Autoscales the preview graph along the y-axis. The default is to autoscale the amplitude.
Displayed signal	Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).
Output Signals	Displays the filtered signal.
Autoscale amplitude	Autoscales the preview graph along the y-axis. The default is to autoscale the amplitude.
Input	The following option applies to the LabVIEW SignalExpress step: • **Input signal**—Specifies the input signal to filter.
Configuration	Contains the following option:

- **Filter Specifications**—Contains the following options:
  - **Mode**—Specifies the mode of filter to use. You can select from the following options:
    - **IIR filter**—Specifies an IIR filter, which is a recursive digital filter with infinite impulse response. IIR filters operate on current and past input values and current and past output values. IIR filters can achieve the same level of attenuation as FIR filters but with fewer coefficients. For this reason, IIR filters can be faster and more efficient than FIR filters.
    - **FIR filter**—Specifies an FIR filter, which is a digital filter with finite impulse response. FIR filters operate only on current and past input values. Because an FIR filter does not depend on past outputs, the impulse response decays to zero in a finite amount of time. Use FIR filters for applications that require linear phase responses.
  - **Type**—Specifies the type of filter to use. You can select from the following options:
    - **Lowpass**—(Default) Passes low frequencies and attenuates high frequencies.
    - **Highpass**—Passes high frequencies and attenuates low frequencies.

- **Bandpass**—Passes a certain band of frequencies. Use the **Low cutoff (Hz)** and the **High cutoff (Hz)** fields to specify the band.
- **Bandstop**—Attenuates a certain band of frequencies. Use the **Low cutoff (Hz)** and the **High cutoff (Hz)** fields to specify the band.

– **Topology**—[Mode: IIR Filter] Specifies the design type of an IIR filter. You can select from the following options:
  - **Off**—Does not filter the signal.
  - **Butterworth**—(Default) Applies a Butterworth filter to the signal. Butterworth filters have a smooth, monotonically decreasing frequency response.
  - **Chebyshev**—Applies a Chebyshev filter to the signal. Chebyshev filters can achieve a sharper transition between the passband and the stopband with a lower order filter than Butterworth filters.
  - **Inverse Chebyshev**—Applies an Inverse Chebyshev filter to the signal. Inverse Chebyshev filters are similar to Chebyshev filters, but they distribute the error over the stopband instead of the passband and are maximally flat in the passband instead of the stopband.
  - **Elliptic**—Applies an Elliptic filter to the signal. Elliptic filters

minimize the peak error by distributing it over the passband and the stopband. Elliptic filters provide the sharpest transition between the passband and the stopband.

- **Bessel**—Applies a Bessel filter to the signal. Bessel filters have maximally flat response in both magnitude and phase. You can use Bessel filters to reduce nonlinear phase distortion inherent in all IIR filters.

– **Order—**[Mode: IIR filter] Determines the order of an IIR filter, which must be greater than zero. The default is 2. Increasing the value of **Order** causes the transition between the passband and the stopband to become steeper. However, as the value of **Order** increases, the processing speed becomes slower, and the number of distorted points at the start of the signal increases.

– **Number of taps**—[Mode: FIR filter] Specifies the total number of FIR coefficients, which must be greater than zero. The default is 49. Increasing the value of taps causes the transition between the passband and the stopband to become steeper. However, as the value of **Number of taps** increases, the processing speed becomes slower.

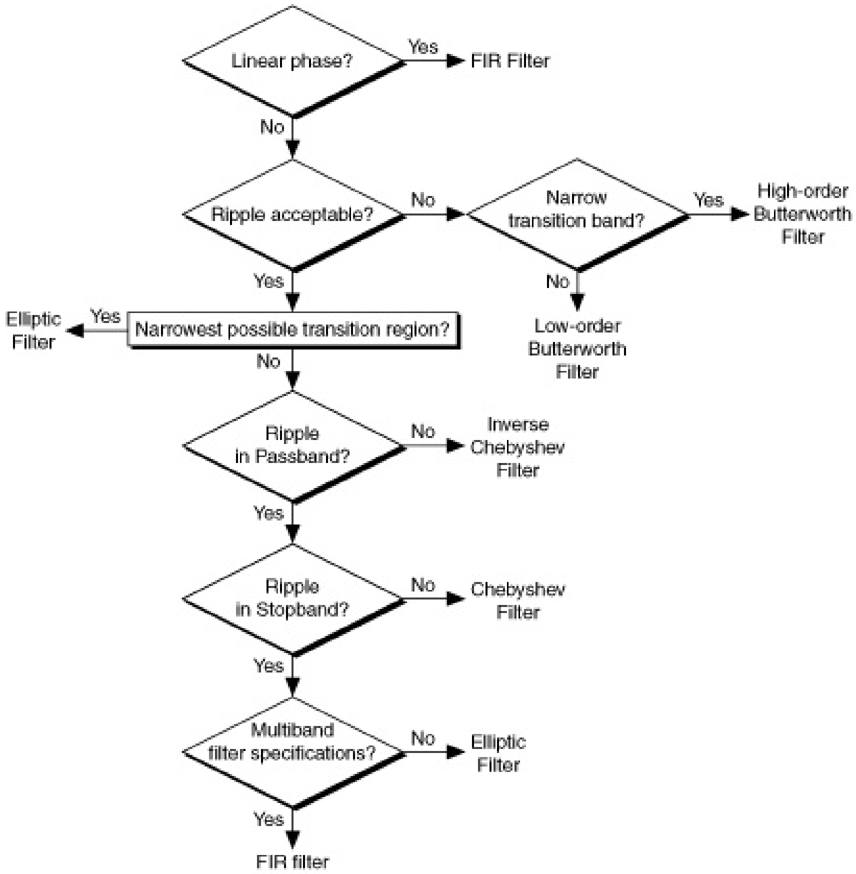
– **Cutoff (Hz)**—[Type: Lowpass, Highpass] Specifies the cutoff frequency of the filter when you select a **Lowpass** or **Highpass** filter type. The default is 100 Hz.

– **Low cutoff (Hz)**—[Type: Bandpass,

		Bandstop] Specifies the lower cutoff frequency when you select a **Bandpass** or **Bandstop** filter type. The default is 100 Hz. – **High cutoff (Hz)**—[Type: Bandpass, Bandstop] Specifies the higher cutoff frequency when you select a **Bandpass** or **Bandstop** filter type. **High cutoff (Hz)** must be greater than **Low cutoff (Hz)** and observe the Nyquist criterion. The default is 200 Hz. – **Filter Magnitude Response (dB)**—Displays the magnitude response of the filter you specify.

Filter Details

The best filter **Mode**, **Type**, and **Topology** to use depends on the analysis you want to perform. Use the following illustration as a guideline for selecting the appropriate filter for an analysis project:



Scaling and Conversion (Time Domain)

Performs gain and offset scaling on a time signal, gain scaling on a frequency-domain magnitude signal, or time-delay correction on a frequency-domain phase signal.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Input Signal</td><td>Displays the input time signal to be scaled.
Displayed signal—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input.
Note If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).</td></tr>
<tr><td>Scaled Signal</td><td>Displays the scaled time signal.</td></tr>
<tr><td>Input</td><td>Contains the following option:
Input signal—Specifies the analog waveform for the step.</td></tr>
<tr><td>Configuration</td><td>Contains the following option:
Operation Setup—Contains the following options:
Pre-gain offset—Specifies the amount of offset to add to the signal before gain. The default is 0.
Gain—Specifies the multiplication factor to apply to the time signal. The default is 1.
Post-gain offset—Specifies the amount of offset to add to the signal after gain. The default is 0.
Equivalent diagram—Displays the LabVIEW equivalent block diagram of the scaling operation you select.</td></tr>
</table>

Scaling and Conversion (Frequency Domain)

Performs gain and offset scaling on a time signal, gain scaling on a frequency-domain magnitude signal, or time-delay correction on a frequency-domain phase signal.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Input Signal</td><td>Displays the input signal.
Displayed signal—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input.
Note If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).</td></tr>
<tr><td>Scaled Signal</td><td>Displays the scaled signal.</td></tr>
<tr><td>Input</td><td>Contains the following option:
Input signal—Specifies the analog waveform for the step.</td></tr>
<tr><td>Configuration</td><td>Contains the following option:
Operation Setup—Contains the following options:
Gain—Specifies the gain to apply to the signal.
Gain representation—Specifies if the gain value is represented in decibels.
Equivalent diagram—Displays the LabVIEW equivalent block diagram of the scaling operation you select.
Correction delay—Specifies the delay value to use to correct the phase signal.
Output in degrees—Specifies if the scaled phase signal is represented in radian or in degrees.
Unwrap phase—Specifies if the phase</td></tr>
</table>

	of the output signal is unwrapped.

Subset and Resample (Frequency Domain)

Extracts a subset of an input signal using the **Start frequency** and **Subset length** you specify or resamples an input signal using the frequency interval (**df**) you specify. If you specify a larger **df**, Subset and Resample downsamples the signal. If you specify a smaller **df**, Subset and Resample upsamples the signal.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Input Signal</td><td>Displays the input frequency-domain signal to be scaled.
Displayed signal—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input.
Note If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).</td></tr>
<tr><td>Processed Signal</td><td>Displays the processed signal.</td></tr>
<tr><td>Input</td><td>Contains the following option:
Input signal—Specifies the analog waveform for the step.</td></tr>
<tr><td>Configuration</td><td>Contains the following options:
Subset Setup—Contains the following options:
Extract subset—Specifies whether to extract a frequency subset of the signal using the Start frequency and Subset length you specify.
Start frequency—Specifies the start frequency of the subset signal.
Subset length—Specifies the frequency span of the subset signal.
Resampling Setup—Contains the following options:
Resample—Resamples the entire signal</td></tr>
</table>

	or subset you select. – **Unwrap phase**—Specifies if the phase of the output signal is unwrapped. – **Open interval**—Specifies if the subset is an open or closed interval. For example, if an input waveform contains 3 data elements at t={0, dt, 2dt}, an open interval defines the waveform as extending over the time interval $0 \le t < 2dt$, and a closed interval defines the waveform as extending over the time interval $0 \le t < 3dt$. Place a checkmark in this checkbox to specify an open interval. – **Interpolation type**—Contains the following options: • **Coerce**—Sets each output sample value to equal the input sample value that is closest to it in frequency. • **Linear**—Sets each output sample value to be a linear interpolation between the two input samples that are closest to it in frequency. • **Spline**—Uses the spline interpolation algorithm to compute the resampled values. • **Filter based**—Uses an interpolation method based on the convolution of the signal with a finite impulse response (FIR) filter. – **df**—Specifies the frequency resolution of the resampled frequency-domain signal. The default is 0.
Filter Setup	Available when you select **Filter based** from **Interpolation type**. Contains the following option:

	• **FIR Filter Specification**—Contains the following options: – **Normalized bandwidth**—The normalized cut-off frequency of the FIR filter to use. The default is 0.4000. – **Alias rejection (dB)**—The minimum stopband attenuation of the FIR filter to use. The default is 80 dB.

Subset and Resample (Time Domain)

Extracts a subset of an input signal using the **Start position** and **Subset length** you specify or resamples an input signal using the time interval (**dt**) you specify. If you specify a larger **dt**, Subset and Resample downsamples the signal. If you specify a smaller **dt**, Subset and Resample upsamples the signal.

Parameter	Description
Input Signal	Displays the input signal. • **Displayed signal**—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).
Processed Signal	Displays the processed signal.
Input/Output	Contains the following options: • **Input**—Contains the following option: – **Input signal**—Specifies the analog waveform for the step. • **Output**—Contains the following options: – **Export pre-subset signal**—Adds the signal that precedes the subset as an output of Subset and Resample. – **Export post-subset signal**—Adds the signal that follows the subset as an output of Subset and Resample.
Configuration	Contains the following options: • **Subset Setup**—Contains the following options: – **Extract subset**—Specifies whether to extract a time subset of the input signal using the **Start position** and **Subset**

length you specify.

  - **Relative time**—Specifies if the **Start position** value is an absolute timestamp value or a time offset relative to the first sample of the input signal.
  - **Start position**—Specifies the start position of the subset signal.
  - **Subset length**—Specifies the time span of the subset signal.

- **Resampling Setup**—Contains the following options:
  - **Resample**—Resamples the entire signal or subset you select.
  - **Optim. for single record**—Optimizes the resampling operation for a single record. If you remove the checkmark from this checkbox, the resampling process assumes that the signals are continuous until you reset the signals.
  - **Open interval**—Specifies if the subset is an open or closed interval. For example, if an input waveform contains 3 data elements at t={0, dt, 2dt}, an open interval defines the waveform as extending over the time interval $0 \le t < 2dt$, and a closed interval defines the waveform as extending over the time interval $0 \le t < 3dt$. Place a checkmark in this checkbox to specify an open interval.
  - **Interpolation type**—Contains the following options:
    - **Coerce**—Sets each output sample value to equal the input sample value that is closest to it in time.
    - **Linear**—Sets each output sample value to be a linear

	interpolation between the two input samples that are closest to it in time. • **Spline**—Uses the spline interpolation algorithm to compute the resampled values. • **Filter based**—Uses an interpolation method based on the convolution of the signal with a finite impulse response (FIR) filter. – **dt—**Specifies the time resolution of the resampled time-domain signal. The default is 0.
Filter Setup	Available when you select **Filter based** from **Interpolation type**. Contains the following option: • **FIR Filter Specification**—Contains the following options: – **Normalized bandwidth**—The normalized cut-off frequency of the FIR filter to use. The default is 0.4000. – **Alias rejection (dB)**—The minimum stopband attenuation of the FIR filter to use. The default is 80 dB.

Time Averaging

Performs time averaging on a time signal or scalar input.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Input Signal</td><td>Displays the input signal.
Displayed signal—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input.
Note If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).</td></tr>
<tr><td>Autoscale amplitude</td><td>Autoscales the preview graph along the y-axis. The default is to autoscale the amplitude.</td></tr>
<tr><td>Averaged Signal</td><td>Displays the averaged time signal.</td></tr>
<tr><td>Autoscale amplitude</td><td>Autoscale amplitude—Autoscales the preview graph along the y-axis. The default is to autoscale the amplitude.</td></tr>
<tr><td>Configuration</td><td>Contains the following options:
Input signal—Specifies the input signal for the step.
Averaging Configuration—Contains the following options:
Weighting mode—Specifies a linear or exponential weighting mode for the time averaging operation. The default is Exponential.
Number of avg.—Specifies the number of averages to use for time averaging.
Averaging mode—Sets the averaging mode. Contains the following options:
Running avg.—Specifies to calculate a running average of all</td></tr>
</table>

<table>
<tr><td></td><td>
input signals for the duration of the operation.
Block avg. (auto-restart)—Restarts the averaging process as soon as Avg. count. reaches the value you specify in Number of avg.
Status—Contains the following options:
Data ready—Indicates when the averaging process is done and the averaged data are ready.
Avg. counter—Displays the averaging progress.
</td></tr>
<tr><td>Advanced Measurement Setup</td><td>Contains the following option:
Only return data when ready—Specifies to only return an averaged signal when LabVIEW SignalExpress has processed the number of signals specified in the Number of avg. field.
</td></tr>
</table>

Window

Applies the window you select to the time-domain signal.

Parameter	Description
Input and Output Signals	Displays the input signal and the windowed signal. • **Displayed signal**—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).
Input signal	Specifies the analog waveform for the step.
Window Setup	Contains the following option: • **Window**—Specifies the window to apply to the input signal. Options include None, Hanning, Hamming, Blackman-Harris, Exact Blackman, Blackman, Flat Top, 4 Term B-Harris, 7 Term B-Harris, Low Sidelobe, and Gaussian. The default is None.
Window Information	Contains the following results: • **Coherent gain**—Indicates the coherent gain of the window you select. • **Eq. noise bandwidth**—Indicates the equivalent noise bandwidth of the window you select.

Arithmetic (Frequency Domain)

Performs arithmetic operations on two signals. The available operations change depending on the type of the input signals you select.

To select the correct type of operation, select the first signal to process from the **Input signal 1** pull-down menu. The **Input signal 2** pull-down menu displays only the list of compatible signals, and the step displays the available operations.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Input Signals</td><td>Displays the two input signals.
Displayed signal—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input.
Note If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).</td></tr>
<tr><td>Resulting Signal</td><td>Displays the signal that results from the arithmetic operation.</td></tr>
<tr><td>Input</td><td>Contains the following options:
Input signal 1—Specifies the first input signal.
Input signal 2—Specifies the second input signal.</td></tr>
<tr><td>Configuration</td><td>Contains the following option:
Operation Setup—Contains the following options:
Operation—Specifies to add, subtract, multiply, or divide the signals after alignment. The Resulting Signal graph displays the result of the operation. The default is Subtract.
Output unit—Specifies to represent the result of a magnitude operation in decibels or to represent the result of a</td></tr>
</table>

<table>
<tr>
<td></td>
<td>

phase operation in degrees or radians.
Interpolate if needed—Resamples the signals to align the frequency bins.
Interpolation type—Contains the following options:

Coerce—Sets each output sample value to equal the input sample value that is closest to it in frequency.
Linear—Sets each output sample value to be a linear interpolation between the two input samples that are closest to it in frequency.
Spline—Uses the spline interpolation algorithm to compute the resampled values.
Filter based—Uses an interpolation method based on the convolution of the signal with a finite impulse response (FIR) filter.

Resulting interval—Specifies if the resulting signal covers the Common or Global frequency span of the signals. The default is Global.

</td>
</tr>
<tr>
<td>Filter Setup</td>
<td>Available when you select Filter based from Interpolation type. Contains the following option:

FIR Filter Specification—Contains the following options:

Normalized bandwidth—The normalized cut-off frequency of the FIR filter to use. The default is 0.4000.
Alias rejection (dB)—The minimum stopband attenuation of the FIR filter to use. The default is 80 dB.

</td>
</tr>
</table>

Arithmetic (Time Domain)

Performs arithmetic operations on two signals. The available operations change depending on the type of the input signals you select.

To select the correct type of operation, select the first signal to process from the **Input signal 1** pull-down menu. The **Input signal 2** pull-down menu displays only the list of compatible signals, and the step displays the available operations.

Parameter	Description
Input Signals	Displays the two input signals. • **Displayed signal**—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).
Resulting Signal	Displays the signal that results from the arithmetic operation.
Input	Contains the following options: • **Input signal 1**—Specifies the first input signal. • **Input signal 2**—Specifies the second input signal.
Configuration	Contains the following option: • **Operation Setup**—Contains the following options: – **Operation**—Specifies to add, subtract, multiply, or divide the signals or compute the RMS sum of the signals. The default is to add the signals. – **Ignore timestamp**—Ignores eventual differences in timestamps by forcing the second signal timestamp to be equal to the first signal timestamp.

	- **Interpolate if needed**—Resamples the signals to align the samples.
	- **Optimized for single record**—Optimizes the resampling operation for executing Arithmetic once. Place a checkmark in this checkbox to speed execution when Arithmetic runs once.
	- **Interpolation type**—Contains the following options:
	  • **Coerce**—Sets each output sample value to equal the input sample value that is closest to it in time.
	  • **Linear**—Sets each output sample value to be a linear interpolation between the two input samples that are closest to it in time.
	  • **Spline**—Uses the spline interpolation algorithm to compute the resampled values.
	  • **Filter based**—Uses an interpolation method based on the convolution of the signal with a finite impulse response (FIR) filter.
	- **Resulting interval**—Specifies if the resulting signal covers the **Common** or **Global** time interval of the signals.

Filter Setup	Available when you select **Filter based** from **Interpolation type**. Contains the following option: • **FIR Filter Specification**—Contains the following options: - **Normalized bandwidth**—The normalized cut-off frequency of the FIR filter to use. The default is 0.4000.

		– **Alias rejection (dB)**—The minimum stopband attenuation of the FIR filter to use. The default is 80 dB.

Formula

Performs math operations on up to four input variables. By default, Formula processes only one variable, but you can place checkmarks in the **Enable** checkboxes to enable more variables. Enter a formula in the **Formula** field using the variable names you specify in the **Alias** fields to represent the **Input variable** values.

You can use time waveform or scalar values for the **Input variable** values. If you apply the **Formula** to multiple time waveforms, the waveforms must be the same size or Formula returns an error.

If all the **Input variable** values are time waveforms, Formula returns a time waveform. If all the **Input variable** values are scalar values, Formula returns a scalar value. If the **Input variable** values are a mix of time waveforms and scalar values, Formula returns a time waveform.

Details

Parameter	Description
Input and Output Waveforms	Displays the waveforms you specify in the **Input variable** fields and the processed data that Formula returns after you apply the **Formula** to the waveforms. This graph appears only when you select a waveform as an **Input variable**.
Input and Output Scalars	Displays the scalar values you specify in the **Input variable** fields and the processed data that Formula returns when you apply the **Formula** to the values. This chart appears only when you select a scalar value as an **Input variable**.
Input variable 0	Specifies the first variable to use.
Alias 0	Specifies an alias name for the first variable.
Enable 1	Enables a second variable.
Input variable 1	Selects the second variable to use.
Alias 1	Specifies an alias name for the second variable.
Enable 2	Enables a third variable.
Input	Selects the third variable to use.

variable 2	
Alias 2	Specifies an alias name for the third variable.
Enable 3	Enables a fourth variable.
Input variable 3	Selects the fourth variable to use.
Alias 3	Specifies an alias name for the fourth variable.
Operation Setup	Contains the following options: • **Formula**—Specifies the formula for the computation. • **Valid**—Reports if the formula is valid or invalid. • **Ignore timestamps**—Specifies to ignore timestamps on input variables that are time waveforms. Place a checkmark in this checkbox to use time waveforms in the **Formula** that have different timestamps.

Formula Details

If you specify multiple waveforms for the **Input variable** values, LabVIEW SignalExpress applies the **Formula** to each sample of the waveforms, which is why multiple waveforms must be the same size. If you specify a mix of waveforms and scalar values for the **Input variable** values, LabVIEW SignalExpress applies the **Formula** to the scalar value and each sample of the waveform. For example, if you specify a **Formula** that multiplies a waveform and a scalar value, LabVIEW SignalExpress multiplies each sample of the waveform by the scalar value.

The following table displays the math functions Formula supports.

Function	Description
abs(*x*)	Returns the absolute value of *x*.
acos(*x*)	Computes the inverse cosine of *x* in radians.
acosh(*x*)	Computes the inverse hyperbolic cosine of *x*.
asin(*x*)	Computes the inverse sine of *x* in radians.
asinh(*x*)	Computes the inverse hyperbolic sine of *x*.
atan(*x*)	Computes the inverse tangent of *x* in radians.
atanh(*x*)	Computes the inverse hyperbolic tangent of *x*.
ceil(*x*)	Rounds *x* to the next higher integer (smallest integer ≤ *x*).
ci(*x*)	Evaluates the cosine integral for any real nonnegative number *x*.
cos(*x*)	Computes the cosine of *x*, where *x* is in radians.
cosh(*x*)	Computes the hyperbolic cosine of *x*.
cot(*x*)	Computes the cotangent of *x* (1/tan(*x*)), where *x* is in radians.
csc(*x*)	Computes the cosecant of *x* (1/sin(*x*)), where *x* is in radians.
exp(*x*)	Computes the value of *e* raised to the *x* power.
expm1(*x*)	Computes one less than the value of *e* raised to the *x* power ((*e*^*x*)–1).
floor(*x*)	Truncates *x* to the next lower integer (largest integer ≤ *x*).
getexp(*x*)	Returns the exponent of *x*.
gamma(*x*)	Evaluates the gamma function or incomplete gamma function

	for *x*.
getman(*x*)	Returns the mantissa of *x*.
int(*x*)	Rounds *x* to the nearest integer.
intrz(*x*)	Rounds *x* to the nearest integer between *x* and zero.
ln(*x*)	Computes the natural logarithm of *x* (to the base of *e*).
lnp1(*x*)	Computes the natural logarithm of (*x* + 1).
log(*x*)	Computes the logarithm of *x* (to the base of 10).
log2(*x*)	Computes the logarithm of *x* (to the base of 2).
rand( )	Produces a floating-point number between 0 and 1 exclusively.
si(*x*)	Evaluates the sine integral for an real number *x*.
sec(*x*)	Computes the secant of *x*, where *x* is in radians (1/cos(*x*)).
sign(*x*)	Returns 1 if *x* is greater than 0, returns 0 if *x* is equal to 0, and returns –1 if *x* is less than 0.
sin(*x*)	Computes the sine of *x*, where *x* is in radians.
sinc(*x*)	Computes the sine of *x* divided by *x* (sin(*x*)/x), where *x* is in radians.
sinh(*x*)	Computes the hyperbolic since of *x*.
spike(*x*)	Generates the spike function for any real number *x*.
sqrt(*x*)	Computes the square root of *x*.
step(*x*)	Generates the step function for any real number *x*.
tan(*x*)	Computes the tangent of *x*, where *x* is in radians.
tanh(*x*)	Computes the hyperbolic tangent of *x*.

Interactive Alignment

Aligns two plots so you can compare them. You can align the **Test signal in** signal with the **Ref. signal in** signal manually by dragging and/or expanding the **Test** plot on the graph or by using algorithms to automatically align steps, pulses, or periodic parameters.

Details

Parameter	Description
Input Signals	Displays the two signals to align.
Autoscale Y(x)	Adjusts the vertical scale to reflect the data from the input signals.
Comparison Result Signal	Displays the comparison signal that results from the operation you specified with **Operation** in the **Resampling and Comparison Setup** section of the **Resampling** page.
Autoscale Comparison Signal	Adjusts the vertical scale to reflect the result of the operation on the two aligned signals.
Autoscale x	Adjusts the time scale to reflect the data to display.
Input/Output	Contains the following options: • **Ref. signal in**—Specifies the reference input signal. • **Test signal in**—Specifies the test input signal to align with the reference signal. • **Export aligned signals**—Exports the **Ref. signal in** and **Test signal in** signals to the Project View. The Interactive Alignment step resamples the **Test signal in** to match the **Ref. signal in** timing parameters. • **Export x-offset result**—Exports the x-offset value the **Geometry Parameters** section of the **Alignment** page displays to the Project View. • **Export y-offset result**—Exports the y-offset value the **Geometry Parameters** section of the

	Alignment page displays to the Project View. • **Export x-gain result**—Exports the x-gain value the **Geometry Parameters** section of the **Alignment** page displays to the Project View. • **Export y-gain result**—Exports the y-gain value the **Geometry Parameters** section of the **Alignment** page displays to the Project View.
Alignment	Contains the following options: • **Geometry Parameters**—Contains the following options: – **x-offset**—Sets or returns the time shift (offset) of the alignment operation. – **y-offset**—Sets or returns the amplitude offset of the alignment operation. – **x-gain**—Sets or returns the time stretch (gain) of the alignment operation. – **y-gain**—Sets or returns the amplitude gain of the alignment operation. – **Ignore x0**—Forces the timestamp value of the test signal to equal the timestamp value of the reference signal. – **Allow x-offset**—Allows manual time shift (offset) of the test signal. – **Allow y-offset**—Allows manual amplitude offset of the test signal. – **Allow x-gain**—Allows manual time stretch (gain) of the test signal. – **Allow y-gain**—Allows manual amplitude gain of the test signal. • **Alignment Conditions**—Contains the following options: – **Mode**—Specifies the mode Interactive Alignment uses to align the signals. You can select from the following options: • **Manual**—Allows manual alignment of the test signal.

- **Auto-Impulse**—Selects an automatic alignment algorithm based on the assumption that the signals include a positive or negative impulse pattern.
- **Auto-Step**—Selects an automatic alignment algorithm based on the assumption that the signals include a rising or falling step pattern.
- **Auto-Periodic**—Selects an automatic alignment algorithm based on the assumption that the signals are periodic.

– **Invert signal**—Inverts the input test signal.

– **Criterion**—Specifies the following alignment criterion:

- **Align Base and Peak**—Aligns the two impulses to align the base and peak levels and to align the peak positions in time.
- **Align 50-50%**—Aligns the two impulses to superpose their respective 50% rising and falling edge points.
- **Align Edge to User Levels**—Aligns the rising or the falling edge of the impulses to superpose the **Low level (%)** and **High level (%)** points.
- **Align Impulse to User Levels**—Aligns the two impulses to superpose the points the **Rising level (%)** and **Falling level (%)** specify on both plots respectively.

Step criterion:

- **Align Low, High and User**—Aligns the two steps so the low levels (0%) and high levels (100%) are aligned and the points on the edges **Mid level (%)** specifies are superposed.
- **Align 10% and 90%**—Aligns the two steps so the 10% and 90% points on the rising or the falling edges are superposed.
- **Align to User Levels**—Aligns the two steps so the points **Low level (%)** and **High level (%)** specify on the rising or the falling edges are superposed.

Periodic criterion:

- **Align Freq, Phase and p-p**—Aligns the two periodic signals so the fundamental tones are superposed.

– **Falling edge**—Specifies to perform the edge alignment operation on the rising or falling edge of the impulses or steps.

– **Level A**—Contains the following options:

- **Low level (%)**—Specifies the level of a signal point to use as the low reference in an edge alignment operation. The unit is a percentage of the amplitude of the impulse or the step to align. The default is 10.

- **Rising level (%)**—Specifies the level of the rising edge points to superpose in an impulse alignment operation. The default is

<table>
<tr>
<td></td>
<td>

50.
Mid level (%)—Specifies the level of a signal point to use as the medium reference in a step alignment operation. The unit is a percentage of the impulse or the step to align. The default is 50%.

Level B—Contains the following options:

High level (%)—Specifies the level of a signal point to use as the high reference in an edge alignment operation. The unit is a percentage of the amplitude of the impulse or the step to align. The default is 90.
Falling level (%)—Specifies the level of the falling edge points to superpose in an impulse alignment operation. The default is 50.

</td>
</tr>
<tr>
<td>Resampling</td>
<td>Contains the following options:

Resampling and Comparison Setup—Contains the following options:

Interpolation type—Contains the following options:

Coerce—Sets each output sample value to equal the input sample value that is closest to it in time.
Linear—Sets each output sample value to be a linear interpolation between the two input samples that are closest to it in time.
Spline—Uses the spline interpolation algorithm to compute

</td>
</tr>
</table>

	the resampled values. • **Filter based**—Uses an interpolation method based on the convolution of the signal with a finite impulse response (FIR) filter. – **Operation**—Specifies to add, subtract, multiply, or divide the signals after alignment. The **Resulting Signal** graph displays the result of the operation. The default is Subtract. • **FIR Filter Specification**—Contains the following options: – **Normalized bandwidth**—The normalized cut-off frequency of the FIR filter to use. The default is 0.4000. – **Alias rejection (dB)**—The minimum stopband attenuation of the FIR filter to use. The default is 80 dB.

Interactive Alignment Details

Moving the Test plot

You can drag the **Test** plot to move it. When you release the mouse button, the graph performs an autoscale operation to optimize the viewing of the plots unless you remove the checkmarks from the **Autoscale** checkboxes. LabVIEW SignalExpress does not update the lower graph that displays the **Comparison Result Signal** when you drag the **Test** plot, but it performs a new comparison operation as soon as you release the mouse button.

A small cross on the upper graph called the Anchor point marks the location where you last released the mouse. To move the Anchor point position, click on the new location.

Expanding the Test plot

You also can expand the **Test** plot in both directions, corresponding to a gain/attenuation of the signal amplitude in the vertical direction and a time expansion/compression in the horizontal direction. To expand the plot, press the <Alt> key, click the graph, and drag it. The expansion keeps the position of the Anchor point unchanged; and the mouse position point in the plane at the start of the expansion follows the mouse move.

Locking Move or Expansion

You can prevent unwanted moves and/or expansions in specific directions by removing the checkmark from the corresponding **Allow x-gain**, **Allow y-gain**, **Allow x-offset**, or **Allow y-offset** checkbox. Notice that preventing certain moves or expansion conflicts with the actual position of the Anchor point and results in slightly different expansion behaviors.

Alignment Evaluation

You can evaluate the alignment on the lower graph that displays the **Comparison Result Signal**. This signal displays the result of an arithmetic operation you can specify LabVIEW SignalExpress to perform on the two aligned signals. The default is Subtract.

Exporting Alignment Results

You can export the alignment information using the following checkboxes on the **Input/Output** page:

- Export aligned signals
- Export x-offset result
- Export y-offset result
- Export x-gain result
- Export y-gain result

Resampling the Test signal

To align the **Test signal in** with the **Ref. signal in**, you must resample the signal so you can perform a sample by sample arithmetic operation like subtraction. The resampling process ensures that LabVIEW SignalExpress samples the aligned waveforms at the same rate and in phase. You can select different resampling options on the **Resampling** page.

Convert Analog to Digital

Converts an analog waveform to a digital waveform.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Analog signal preview</td><td>Displays the analog waveform you want to convert into a digital waveform.</td></tr>
<tr><td>X scale slider</td><td>Selects the data displayed in the Digital waveform preview.</td></tr>
<tr><td>Signal name preview</td><td>Displays the signal names associated with the signals in the Digital waveform preview.</td></tr>
<tr><td>Digital waveform preview</td><td>Displays the digital waveform converted from the input analog waveform.</td></tr>
<tr><td>Digital preview Y scrollbar</td><td>Scrollbar that allows you to scroll through the Digital waveform preview.</td></tr>
<tr><td>Input</td><td>Contains the following options:
Input signal—Specifies the analog waveform for the step.</td></tr>
<tr><td>Configuration</td><td>Contains the following options:
Analog full scale—The total peak-to-peak range, or the difference between the minimum and maximum, for the analog waveform. For example, if the maximum range of a waveform is 1 and the minimum is -1, the full-scale range for the waveform is 2.
Resolution (bits)—Specifies the number of bits represented in the digital waveform. LabVIEW SignalExpress supports a maximum resolution of 32 bits.
Digital data format—Specifies which binary representation you want to use for the digital data.
Unsigned binary—The data is converted to unsigned binary.</td></tr>
</table>

	– **Offset binary**—The largest negative value (negative full-scale) is represented by all zeros, and the largest positive value (positive full-scale) is represented by all ones. Zero-scale is represented by a one (MSB) followed by all zeros, for example, binary 1000. – **2's complement**—Uses two's complement format, which is a common format for representing signed binary values. This format is similar to Offset Binary, but the MSB is inverted. • **Dithering enabled**—Specifies whether the analog waveform can be dithered. Dithering a waveform adds Gaussian noise to an analog input signal to increase resolution.
Signal Names	Contains the following options: • **Signal names table**—Allows you to specify custom names for your signals. • **Reset to default**—Specifies whether the Signal names are reset to their default states.

Convert Digital to Analog

Converts an input digital waveform into an analog waveform.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Digital waveform preview</td><td>Displays the input digital waveform.</td></tr>
<tr><td>Digital preview Y scrollbar</td><td>Scrollbar that allows you to scroll through the Digital waveform preview.</td></tr>
<tr><td>Signal name preview</td><td>Displays the signal names associated with the signals in the Digital waveform preview.</td></tr>
<tr><td>Analog signal preview</td><td>Displays a preview of the analog waveform.</td></tr>
<tr><td>X scale slider</td><td>Selects the data displayed in the Digital waveform preview.</td></tr>
<tr><td>Input</td><td>Contains the following options:
Input signal—Specifies the digital waveform to convert to an analog waveform.</td></tr>
<tr><td>Configuration and Results</td><td>Contains the following options:
Analog full scale—The total peak-to-peak range, or the difference between the minimum and maximum, for the analog waveform. For example, if the maximum range of a waveform is 1 and the minimum is -1, the full-scale range for the waveform is 2.
Digital data format—Specifies which binary representation you want to use for the digital data.
Unsigned binary—The data is converted to unsigned binary.
Offset binary—The largest negative value (negative full-scale) is represented by all zeros, and the largest positive value (positive full-scale) is represented</td></tr>
</table>

	by all ones. Zero-scale is represented by a one (MSB) followed by all zeros, for example, binary 1000. – **2's complement**—Uses two's complement format, which is a common format for representing signed binary values. This format is similar to Offset Binary, but the MSB is inverted. • **Output signal unit**—Specifies the unit type for the output signal. Selecting **Unitless** specifies to not associate any unit with the output signal. Selecting **Custom** specifies that the output signal contains an associated unit within the data. • **Resolution (bits)**—Returns the number of bits in the converted digital waveform. • **Export resolution**—Exports the resolution to the Project View. • **Y axis unit**—Specifies the unit type for the input signal. Selecting **Unitless** specifies to not associate any unit with the input signal. Selecting **Custom** specifies that the input signal contains an associated unit within the data. • **X axis unit**—Specifies the unit type for the input signal. Selecting **Unitless** specifies to not associate any unit with the input signal. Selecting **Custom** specifies that the input signal contains an associated unit within the data.

Analysis

Use the Analysis steps to analyze your LabVIEW SignalExpress measurements.

 To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Time-Domain Measurements

Use the Time-Domain Measurements steps to perform time domain analysis. The Time-Domain Measurements steps implement some operations commonly used in signal processing.

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Amplitude and Levels

Measures DC, RMS, positive and negative peak, and peak-to-peak values of a signal.

You can measure DC and RMS values with linear or exponential averaging. If you select **Linear**, you can apply a window to the signal.

You also can individually export the different measurement results.

Parameter	Description
Input Signal	Displays the input signal. • **Displayed signal**—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).
Autoscale amplitude	Autoscales the preview graph along the y-axis. The default is to autoscale the amplitude.
Input/Output	Contains the following options: • **Input signal**—Specifies the input signal to be measured. • **Export DC value**—Exports the DC value to the Project View. • **Export RMS value**—Exports the RMS value to the Project View. • **Export +Peak value**—Exports the positive peak value to the Project View. • **Export -Peak value**—Exports the negative peak value to the Project View. • **Export Peak-Peak value**—Exports the difference between the positive and negative peak values to the Project View.
Configuration	Contains the following options:

- **DC-RMS Setup**—Contains the following options:
  - **Averaging type**—Sets the type of averaging to linear or exponential. The default is Linear.
  - **Window**—Sets the type of window when you use linear averaging. Windowing can sometimes help increase measurement accuracy for signals that are dominated by periodic components

    Window is not available if you select Exponential as the averaging type. **Window** contains the following options: Rectangular (none), Hanning, and Low side lobe. The default is Rectangular (none).
- **Peak Setup**—Contains the following option:
  - **Hold peaks**—Specifies to hold the peak levels until you click the **Reset Amplitude and Levels** button or the **Reset All** button or restart your measurement. The default is to not hold the peak levels.
- **DC-RMS Results**—Contains the following options:
  - **DC value**—Returns the measured DC value.
  - **RMS value**—Returns the measured RMS value.
- **Peak Results**—Contains the following options:
  - **+Peak value**—Returns the positive peak value of the input signal.
  - **-Peak value**—Returns the negative peak value of the input signal.
  - **Peak-peak value**—Returns the difference between the positive and negative peak values of the input signal.

Histogram

Calculates the discrete histogram of the input signal. The **Histogram** result runs continuously and accumulates the data from all incoming signals until you click the **Reset Histogram** button on the title bar of the **Histogram** configuration view or you change a configuration parameter. You can compute the resulting bin values as an absolute number of occurrence or as a percentage of the total number of occurrences. You also can display the accumulated bin values in a logarithmic scale. Click the **Auto-config** button below the **Histogram** graph to display standard start-up configuration options.

Parameter	Description
Input Signal	Displays the input signal. • **Displayed signal**—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).
Histogram	Displays the histogram of the input signal.
Auto-config.	Sets the values in **Number of bins**, **Minimum value**, and **Maximum value** based on the input signal.
Log. bin values	Scales the bin values axis to logarithmic scale. The default is to not scale the bin values axis to logarithmic scale.
Input	Contains the following option: • **Input signal**—Specifies the analog waveform for the step.
Configuration	Contains the following option: • **Histogram Specifications**—Contains the following options: – **Number of bins**—Specifies the number of bins. The default is 20.

		– **Minimum value**—Specifies the minimum value. The default is –1. – **Maximum value**—Specifies the maximum value. The default is 1. – **Bin value in percent**—Configures the histogram result to scale by percent. The default is to scale the result by percent. – **Calculation enabled**—Enables calculation of the histogram. The default is to enable the calculation.

Statistics

Performs statistical calculations on time-domain, scalar, or array of scalar data. You can select up to six statistical measurements to perform on your data, and Statistics creates a scalar output for each specified measurement. If the input signal is a waveform, by default Statistics returns a statistical measurement on the current input signal. For scalar data, the Statistics step returns a statistical measurement that represents the entire signal history since you started the project or you reset the step.

Parameter	Description
Input Signal	Displays the input signal. If you wire data to the Express VI and run it, **Input Signal** displays real data. If you close and reopen the Express VI, **Input Signal** displays sample data until you run the Express VI again. • **Displayed signal**—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s). • **Number of points in the data**—Specifies the number of points in the data. The default is 4400.
Input	Contains the following option: • **Input signal**—Specifies the input value.
Configuration And Results	Contains the following options: • **Max**—Specifies to output the maximum value of the current input signal. • **Min**—Specifies to output the minimum value of the current input signal. • **Mean**—Specifies to output the mean value of the current input signal. • **Nb of samples**—Specifies to output the number

	of input samples the Statistics step performs statistical measurements on. • **Combine channels**—Specifies whether Statistics returns single outputs for measurements on groups of input signals. For example, if you configure Statistics to measure the maximum value of the signals, placing a checkmark in this checkbox returns the maximum value of all the samples of all the channels, instead of one maximum value per channel. This option appears only when **Input signal** is a group of signals. • **Standard deviation**—Specifies to output the standard deviation of the current input signal. • **Variance**—Specifies to output the variance of the current input signal. • **Sum**—Specifies to output the sum of the current input signal. • **Measurement duration (s)**—Indicates the duration of each measurement the Statistics step is returning. • **Restart measurement on each iteration**—Specifies to restart the statistical measurement on each iteration of the input signal. If you do not select this option, output statistics represent the cumulative values of the entire input signal.

Timing and Transition

Measures timing and transition parameters on single pulses and on rising and falling edges.

If the signal includes a single positive or negative pulse or a pulse train, you can measure the pulse frequency, period (1/frequency), duration, and duty cycle. You also can use this step to measure the transition time, rise or fall time, the amount of undershoot and overshoot, and the slew rate on rising and falling edges. You can select the pulse polarity and the pulse and/or edge numbers on the **Advanced** page.

Details

Parameter	Description
Input Signal	Displays the input signal to measure. Cross-hair cursors indicate the pulse and edges to use for the measurements. This step marks edges at the pulse and mid-transition points at its center position using the color code in the three result tables. • **Displayed signal**—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).
Autoscale amplitude	Scales the Amplitude (V) axis of the **Acquired Data** graph. The default is to autoscale the amplitude.
Input	Contains the following option: • **Input signal**—Specifies the analog waveform for the step.
Configuration and Results	Contains the following options: • **Pulse results**—Contains the results of the pulse measurements. If one or more measurements you select fail, this step highlights the background color of the failing measurements in

red. To disable failing measurements, remove the checkmark from the corresponding checkboxes. Contains the following options:

  - **Export frequency**—Measures the frequency of a pulse train and exports the measurement result to the Project View.
  - **Export period**—Measures the period of a pulse train and exports the measurement result to the Project View.
  - **Export duration**—Measures the duration of a pulse and exports the measurement result to the Project View.
  - **Export duty cycle**—Measures the duty cycle of a pulse and exports the measurement result to the Project View.

- **Rising edge results**—Contains the results of the rising edge transition measurements. If one or more measurements you select fail, this step highlights the background color of the failing measurements in red. To disable failing measurements, remove the checkmark from the corresponding checkboxes. Contains the following options:
  - **Export rise time**—Measures the rising transition time (rise time) of an edge and exports the measurement result to the Project View. The rise time is the time it takes the signal to change from a low reference level (10% of the amplitude of the signal) to a high reference level (90% of the amplitude of the signal).
  - **Export rising undershoot**—Measures the percentage amount of undershoot that precedes a rising edge and exports the measurement result to the Project View.

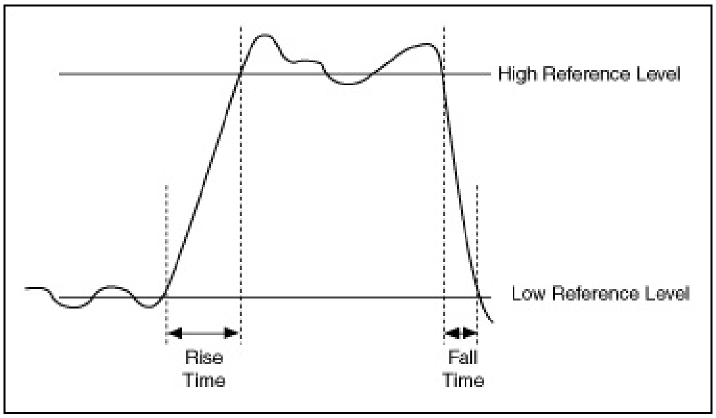
- **Export rising overshoot**—Measures the percentage amount of overshoot that follows a rising edge and exports the measurement result to the Project View.
- **Export rising slew rate**—Measures the slew rate, or the ratio between (90% amplitude – 10% amplitude) and the rise time, of a rising edge and exports the measurement result to the Project View.

• **Falling edge results**—Contains the results of the falling edge transition measurements. If one or more measurements you select fail, this step highlights the background color of the failing measurements in red. To disable failing measurements, remove the checkmark from the corresponding checkboxes. Contains the following options:
  - **Export fall time**—Measures the falling transition time (fall time) of an edge and exports the measurement result to the Project View. The fall time is the time it takes the signal to change from a high reference level (90% of the amplitude of the signal) to a low reference level (10% of the amplitude of the signal).
  - **Export falling undershoot**—Measures the percentage amount of undershoot that follows a falling edge and exports the measurement result to the Project View.
  - **Export falling overshoot**—Measures the percentage amount of overshoot that precedes a falling edge and exports the measurement result to the Project View.
  - **Export falling slew rate**—Measures the slew rate, or the ratio between (90% amplitude – 10% amplitude) and the fall time, of a falling edge and exports the

	measurement result to the Project View.
Advanced	Contains the following options: • **Pulse Definition**—Contains the following options: – **Pulse polarity**—Specifies if the pulse to measure is positive (High Pulse) polarity or negative (Low Pulse) polarity. – **Pulse number**—Specifies which pulse number in a pulse train to use for the measurement. • **Transition Definition**—Contains the following options: – **Rising edge number**—Specifies which rising edge to use for the measurement. – **Falling edge number**—Specifies which falling edge to use for the measurement.

Timing and Transition Details

The following image shows a sample pulse. Timing and Transition uses a high reference level of 90% of the amplitude of the signal and a low reference level of 10% of the amplitude of the signal. In a timing and transition measurement, overshoot is the height of a local maximum preceding a rising or falling edge, depending on the **Pulse polarity** you specify. Undershoot is the height of the local minimum preceding a rising or falling edge, depending on the **Pulse polarity** you specify.



Frequency-Domain Measurements

Use the Frequency-Domain Measurements steps to perform signal analysis that require the data to be converted into frequency domain. The Frequency-Domain Measurements steps perform frequency domain transformations and frequency domain analysis.

 To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Power Spectrum

Computes the averaged magnitude spectrum, power spectrum, or power spectral density for a single or multiple channels. This step can return the spectra in root-mean-square, peak, and peak-to-peak units.

Parameter	Description
Graph	**Spectra**—Displays the spectra for all channels. Use the **Zoom** button to zoom in and out of the display. **Signals**—Displays the time-domain signals for all channels. Use the **Zoom** button to zoom in and out of the display.
View	Specifies if the graph displays time domain signals or the computed power spectra.
Autoscale	Automatically adjusts the scales of the graph to display the data.
Input	Contains the following option: • **Input signal**—Specifies the analog waveform for the step.
Configuration	Contains the following options: • **Window**—Specifies the window to apply to the input signal. Choose from one of the following window options: – **None** – **Hanning** – **Hamming** – **Blackman-Harris** – **Exact Blackman** – **Blackman** – **Flat Top** – **4 Term B-Harris** (Four Term Blackman-Harris) – **7 Term B-Harris** (Seven Term Blackman-Harris)

	– **Low Sidelobe** – **Gaussian** • **Scaling**—Specifies the scaling parameter for the step. – **Spectrum type**—Specifies if the spectrum is in units of **Magnitude** or **Power**, where power equals magnitude squared. The default is **Power**. – **Magnitude scale**—Specifies if zoom power is in linear units or in decibels. The default is decibels. – **Peak conversion**—Specifies the peak scaling of the converted spectrum. You can select **RMS** (default), **Peak**, or **Peak to Peak**. – **Spectral density**—Specifies if the spectrum is returned as power spectral density (PSD). The default is **Off**.
Averaging	Specifies the averaging parameters. • **Averaging mode**—Specifies the averaging mode from the following options: – **No Averaging** (Default) – **Vector Averaging** – **RMS Averaging** – **Peak Hold** • **Weighting mode**—Specifies either **Exponential** or **Linear** weighting. Exponential averaging applies more weight to the most recent data, and linear averaging applies equal weighting to all the data. • **Number of averages**—Specifies the number of averages used by the selected **Weighting mode**. • **Auto-restart**—Specifies if the averaging process automatically restarts once the step reaches the **Number of averages** value. When you set

	Weighting mode to linear, use **Auto-restart** to configure averaging to automatically restart when **Averages completed** equals the **Number of averages**. • **Averages completed**—Displays the number of averages completed. • **Averaging done**—Indicates when the number of **Averages completed** equals or exceeds the **Number of averages**. **Averaging done** is always TRUE if the selected **Averaging** mode is **No averaging**.

Distortion

Performs harmonic distortion analysis and/or SINAD measurement on the input signal.

This step returns the fundamental frequency, the percentage of total harmonic distortion, the total harmonic distortion plus noise value, and the SINAD value in decibels.

This step also returns a time-domain waveform and frequency-domain power spectrum for the different components of the signal, such as fundamental signal, residual signal, or harmonics.

Parameter	Description
Exported Signal	Displays the time signal you selected with **Export signals** in the **Measurement Setup** section on the **Configuration** page. • **Displayed signal**—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).
Exported Power Spectrum	Displays the power spectrum of the signal the **Exported Signal** graph displays.
Autoscale magnitude	Scales the magnitude of the exported power spectrum. The default is to autoscale the magnitude.
Input/Output	Contains the following options: • **Input signal**—Specifies the input signal to be measured. • **Export time signal**—Exports the signal to the Project View as **Export Signals (THD)** or **Export Signals (SINAD)** specifies. • **Export power spectrum**—Exports the power spectrum to the Project View as **Export Signals**

	(THD) or **Export Signals (SINAD)** specifies.
Configuration	Contains the following options: • **Measurement Setup**—Contains the following options: – **Distortion types**—Specifies **Harmonic Only**, **SINAD Only**, or **Harmonic and SINAD** distortion. – **Highest harm.**—Specifies the highest harmonic to include in the calculation of the harmonic distortion. The default is 19, so the THD result is based on harmonics 2 to 19 (both included) assuming that the **Exclude aliased harmonics** checkbox does not include a checkmark. If you place a checkmark in the **Exclude aliased harmonics** checkbox, **Highest harm.** includes only the harmonics below the Nyquist frequency. The Nyquist frequency is half the sample rate of the input signal. – **Export signals (THD)**—Specifies which signals to display on the two graphs and to export to the Project View. Options include Input Signal, Fundamental Tone, Residual Signal, Harmonics Only, Noise and Spurs or None (no signal). The default is Input Signal. – **Export signals (SINAD)**—Specifies which signals to display on the two graphs and to export to the Project View. Options include Input Signal, Fundamental Tone, Residual Signal, or None (no signal). The default is Input Signal. • **Measurement Results**—Contains the following options: – **Fund. Frequency**—Returns the

	detected fundamental frequency of the input signal. – **THD (%)**—Returns the measured percentage of total harmonic distortion up to and including the highest harmonic or limited by the Nyquist frequency. – **SINAD (dB)**—Returns the measured sine in noise and distortion value in decibels. – **THD + Noise (%)**—Returns the measured total harmonic distortion plus noise. Notice that this result always includes all harmonics and is therefore independent of the value you specify for **Highest harm**.
Advanced	Contains the following options: • **Approx. fund. freq. (Hz)**—Returns the center frequency to use in the frequency-domain search for the fundamental tone. A negative value specifies to search for the tone with the highest amplitude. The default is–1. • **Exclude aliased harmonics**—Exclude aliased harmonics should be set to TRUE (default) to include only frequencies less than the Nyquist frequency, or half the sampling rate, in the harmonic search. When set to FALSE, this step continues searching the frequency domain beyond Nyquist by assuming that higher frequency components have aliased according to the following equation. Aliased f = Fs – (f modulo Fs) where Fs = 1/dt = sampling rate.

Tone Extraction

Finds the single tone with the highest amplitude or searches a frequency range you specify to find the single tone with the highest amplitude and returns the frequency, amplitude, and absolute phase for the detected tone. The step also can export **Export signals** to the Project View.

Parameter	Description
Exported Time Signal	Displays the time signal that **Export signals** specifies. • **Displayed signal**—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).
Exported Power Spectrum	Displays the power spectrum in decibels that **Exported Time Signal** specifies.
Autoscale magnitude	Scales the magnitude scale of the power spectrum result graph. The default is to autoscale the magnitude.
Input/Output	Contains the following options: • **Input signal**—Specifies the signal from which to extract tonal data. • **Export time signal**—Exports the time signal that **Exported Time Signal** displays to the Project View. • **Export power spectrum**—Exports the power spectrum result that **Exported Power Spectrum** displays to the Project View.
Configuration	Contains the following option: • **Measurement Setup and Results**—Contains the following options: – **Export signals**—Specifies the signal to display on the graphs and optionally

	export to the Project View. Options include None, Input Signal, Extracted Tone, or Residual Signal, which is the input signal minus the extracted single tone. The default is Input Signal. – **Detected frequency**—Returns the frequency of the detected single tone in hertz. – **Detected amplitude**—Returns the amplitude of the detected single tone. – **Detected phase (deg)**—Returns the phase of the detected single tone in degrees.
Advanced	Contains the following option: • **Advanced Measurement Setup**—Contains the following options: – **Approximate frequency (Hz)**—Specifies the center frequency to use in the frequency-domain search for the single tone. A negative value corresponds to search automatically for the tone with the highest amplitude. The default is –1. – **Search range (% of sample rate)**—Specifies the frequency span as a percentage of the sampling rate for the frequency-domain search for the single tone frequency. The default is 0.25%.

Test and Compare

Use the Test and Compare steps to compare an input signal to user-specified limits.

 To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Limit Test

Tests an input signal or value against user-specified limits and returns information on whether the test passed or failed and, in the case of a failure, where it failed. Limit Test accepts time-domain signals, frequency-domain signals, and scalar values as inputs. You can specify either signals or scalar values for the limits, and you can define the limits or use other signals in the project as the limits.

Details

Parameter	Description
View	Specifies how to display the results of the limit test. You can select from the following options: • **Graph**—(Default) Displays the results of the limit test as a graph. • **Results table**—Displays the results of the limit test as a table.
Limit Test	[View: Graph] Displays the result of the limit test operation. The four plots show the original input signal, the points where the limit test operation failed, and the two limit signals, respectively. • **Displayed signal**—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input. **Note** If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).
Limit test results	[View: Results table] Displays each signal in the limit test and whether or not the signal passed the test.
Autoscale y-axis	[View: Graph] Specifies whether to autoscale the y-axis on the **Limit Test** graph.
selected test	Indicates whether the test of the signal displayed on the **Limit Test** graph passed or failed. This indicator appears only when you test a group of signals.

<table>
<tr><td>all tests</td><td>Indicates if the tests passed or failed.</td></tr>
<tr><td>Input</td><td>The following options apply only to the Limit Test step in LabVIEW SignalExpress:

Input signal—Specifies the input signal or scalar value.
Upper limit—[Limits source: Input Signals, Input Scalars] Specifies the upper limit signal or value.
Note When a LabVIEW SignalExpress project runs continuously, LabVIEW SignalExpress reads the Upper limit and Lower limit only on the first iteration of the project. If you select a limit signal that continuously changes, LabVIEW SignalExpress uses only the first iteration of the signal to perform the limit test.
Lower limit—[Limits source: Input Signals, Input Scalars] Specifies the lower limit signal or value.
Limit—[Limits source: Input Signals, Input Scalars AND Limits window based on: Single Limit & Range] Specifies the single limit signal or value.
</td></tr>
<tr><td>Input type</td><td>The following options apply only to the Limit Test Express VI in LabVIEW:

Time Waveform—Specifies to perform limit testing on a time domain signal.
Frequency Waveform—Specifies to perform limit testing on a frequency spectrum.
Scalar—Specifies to perform limit testing on a single scalar value.
</td></tr>
<tr><td>Configuration</td><td>Contains the following options for configuring the limits for the limit test:

Limit Setup—Contains the following options:

Limits source—Specifies the source of the limits for the limit test. You can select from the following options:

</td></tr>
</table>

- **Input Signals**—Uses output signals from previous steps or Express VIs as the limits.
- **Input Scalars**—Uses output scalar values from previous steps or Express VIs as the limits. You specify the scalar values on the **Input** page.
- **User Defined Signals**—Uses signals you define for the limits. If you set **Limits window based on** to **Two Limits**, click the **Define upper limit** and **Define lower limit** buttons to display the **Define Signal** dialog box and interactively define the limit signals. If you set **Limits window based on** to **Single Limit & Range**, click the **Define single limit** button to display the **Define Signal** dialog box and interactively define the limit signal.
- **User Defined Constants**—(Default) Uses constant values that you specify for the limits. If you set **Limits window based on** to **Two Limits**, use the **Upper constant** and **Lower constant** fields to specify the limits. If you set **Limits window based on** to **Single Limit & Range**, use the **Limit constant** field to specify the limit.

– **Compare mode**—Specifies the comparison mode Limit Test uses to compare the input signal to the limits you specify. You can select from the following options:

- **Between Limits**—(Default) Determines whether the input signal is between the limits you specify.
- **Outside Limits**—Determines whether the input signal is outside the limits you specify.
- **>Lower Limit**—Determines whether the input signal is above the lower limit you specify.
- **<Upper Limit**—Determines whether the input signal is below the upper limit you specify.

- **Limits window based on**—Specifies how you define the limits used for the limit test. You can select from the following options:
  - **Two Limits**—Specifies that you define two limits for the limit test.
  - **Single Limit & Range**—Specifies that you define a single limit and a range of gain and offset values for the limit test.
- **Upper constant**—[Limits source: User Defined Constants AND Limits window based on: Two Limits] Specifies the value of the upper limit constant. The default is 1.
- **Lower constant**—[Limits source: User Defined Constants AND Limits window based on: Two Limits] Specifies the value of the lower limit constant. The default is –1.
- **Limit constant**—[Limits source: User Defined Constants AND Limits window based on: Single Limit & Range]

Specifies the single constant value that, in conjunction with the **Relative Range Specs**, defines the limit values.

- **Define upper limit**—[Limits source: User Defined Signals AND Limits window based on: Two Limits] Displays the **Define Signal** dialog box, which you can use to define the upper limit signal interactively.
- **Define lower limit**—[Limits source: User Defined Signals AND Limits window base on: Two Limits] Displays the **Define Signal** dialog box, which you can use to define the lower limit signal interactively.
- **Define single limit**—[Limits source: User Defined Signals AND Limits window base on: Single Limit & Range] Displays the **Define Signal** dialog box, which you can use to define the limit signal interactively.

- **Relative Range Specs**—Contains the following options:
  - **Upper gain**—[Limits window base on: Single Limit & Range] Specifies the gain value to apply to the single limit to calculate the upper limit. The default is 1.1.
  - **Lower gain**—[Limits window base on: Single Limit & Range] Specifies the gain value to apply to the single limit to calculate the lower limit. The default is 900m.
  - **Upper offset**—[Limits window base on: Single Limit & Range] Specifies the offset value to add to the single limit to calculate the upper limit. The default is 0.
  - **Lower offset**—[Limits window base on:

		Single Limit & Range] Specifies the offset value to add to the single limit to calculate the lower limit. The default is 0.
Advanced	Contains the following options: • **Limits Inclusion**—Contains the following options: – **Upper inclusive**—Specifies whether a value that is exactly on or equal to the upper limit passes the limit test. Place a checkmark in this checkbox to pass a value that is on or equal to the upper limit. – **Lower inclusive**—Specifies whether a value that is exactly on or equal to the lower limit passes the limit test. Place a checkmark in this checkbox to pass a value that is on or equal to the lower limit. • **Timing information**—Contains the following options for time-domain and frequency-domain signals: – **Freq. axis is logarithmic**—Sets the display graph frequency axis to logarithmic and, when the **Limits source** is **User Defined Signals**, computes the limit values between the definition points so the resulting segment appears as a straight line in a logarithmic frequency representation. For example, you can use this to create asymptotic limits fitting filter roll-off in decibels per decade. This parameter appears only if the input signal is a frequency-domain signal. – **Ignore timestamp**—Forces the timestamp of the input signal to 0 so you can define the limit signals relative to the beginning of the input signal. This	

<table>
<tr><td></td><td>parameter appears only if the input signal is a time-domain signal.</td></tr>
<tr><td>Actions</td><td>Contains the following options:

Action on failed—Contains the following options:

Stop project after failed occurs—Specifies to stop running the project after the signal fails the limit test the number of times you specify.
times—Specifies the number of times for the signal to pass or fail the limit test before LabVIEW SignalExpress stops the project.
Action on failed—Specifies the action to perform when the signal fails the limit test the number of times you specify.

none—Specifies to take no additional action.
snapshot of inputs—Specifies to take a snapshot of the current inputs of the Limit Test step.
snapshot of all signals in project—Specifies to take a snapshot of all signals in the project.

Action on passed—Contains the following options:

Stop project after passed occurs—Specifies to stop running the project after the signal passes the limit test the number of times you specify.
times—Specifies the number of times for the signal to pass or fail the limit test before LabVIEW SignalExpress stops the project.
Action on passed—Specifies the action

</td></tr>
</table>

		to perform when the signal passes the limit test the number of **times** you specify. • **none**—Specifies to take no additional action. • **snapshot of inputs**—Specifies to take a snapshot of the current inputs of the Limit Test step. • **snapshot of all signals in project**—Specifies to take a snapshot of all signals in the project.

Limit Test Details

Output Signal Types

Limit Test returns a group of signals named **limit test results**. This group of signals contains the following elements:

- **failed signal**—The input signal(s) or value(s). If you graph **failed signal**, the graph includes the input signal, the limits, and the portions of the signal that failed the limit test.
- **upper limit**—The upper limit used to perform the limit test.
- **lower limit**—The lower limit used to perform the limit test.

Limits

The type of limits you can define for the Limit Test depend on the input signal type.

If the input is a time-domain or frequency-domain signal, you can use the following types of limits:

- Output signals of the same type as the input signal—You can use output signals from other LabVIEW SignalExpress steps or LabVIEW VIs as the limits. For example, you can compare a time-domain signal to two other time-domain signals.
- Output scalar values—You can use output scalar values from other LabVIEW SignalExpress steps or LabVIEW VIs as the limits. Limit Test compares the input signal to the scalar values element by element. For example, you can compare a time-domain signal to two measured DC values.
- User-defined signals of the same type as the input signal—You can define limit signals by clicking the **Define Upper Limit**, **Define Lower Limit**, or **Define Single Limit** buttons. The **Define Signal** dialog box appears, and you can define a limit signal based on user-defined points. Use this dialog box to create a limit signal made up of a series of line segments that connect these points.
- User-defined constants—You can define constant values for the limits. Limit Test compares the input signal to the constant values element by element.

If the input is a scalar value, you can use the following types of limits:

- Output scalar values—You can use output scalar values from other LabVIEW SignalExpress steps or LabVIEW VIs as the limits. Limit Test compares the input signal to the scalar values element by element. For example, you can compare a time-domain signal to two measured DC values.
- User-defined constants—You can define constant values for the limits. Limit Test compares the input value to the constant values.

Compare Mode and Limits Inclusion

Limit Test has four compare modes. These modes indicate if a signal or value is between limits, outside limits, greater than a lower limit, or lower than an upper limit. You can choose the exact limit values to include or not include in the test. These limit values result in a failing or passing test where the input value equals the limit value.

Defining a Limit Range from a Single Limit

You can define a set of upper and lower limits from a single limit using the gain and offset scaling parameters. Select the limits based on **Single Limit & Range** to enable the user-defined scaling parameters.

Limits Defined in a Logarithmic Frequency Scale

When you use user-defined signals as limits for a frequency-domain signal, Limit Test defines the limits as a series of line segments that connect user-defined points. By default, Limit Test assumes the frequency axis is linear so a linear relationship exists between the frequency and the magnitude or phase values. The **Freq. axis is logarithmic** checkbox specifies whether to display a logarithmic frequency axis and define the limit signals so the connection between the points appears as straight lines in the logarithmic frequency scale. You can define the limits in a logarithmic frequency scale if you want to test the asymptotic roll-off of a filter, typically a straight line in a decibel versus logarithmic frequency scale.

Digital Compare

Compares a reference and test signal to determine the number of sample errors.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Digital waveform preview</td><td>Displays the two waveforms to be compared.</td></tr>
<tr><td>Signal name preview</td><td>Displays the signal names associated with the signals in the Digital waveform preview.</td></tr>
<tr><td>Digital preview Y scrollbar</td><td>Scrollbar that allows you to scroll through the Digital waveform preview.</td></tr>
<tr><td>Input</td><td>Contains the following options:
Reference waveform—Displays the signal that will serve as the reference for your comparison. Sample errors are produced when the test waveform differs from this waveform.
Test waveform—Displays the signal you want to compare to the reference signal. Sample errors are produced when the test waveform differs from the reference waveform.</td></tr>
<tr><td>Configuration</td><td>Contains the following options:
Comparison Start Position—Contains the following options:
Reference waveform start—Specifies the position of the sample where you want to start the signal comparison.
Test waveform start—Specifies the position of the sample where you want to start the signal comparison.
Compare subset—Specifies whether you want to compare the entire test signal to the reference signal or whether you only want to compare a subset of the samples. Selecting this checkbox</td></tr>
</table>

	means that a subset that is the number of samples specified in **Number Samples** will be compared, starting at the reference waveform start position you specify. • **Number of samples**—Specifies the number of samples in the subset the VI compares. • **Passed**—Displays the result of the waveform compare. If all specified samples match, the compare passes. Otherwise, the compare fails, and the number of sample errors is returned in **Number of Sample Errors**. • **Number of Sample Errors**—Returns the number of sample errors in the test signal.

Execution Control

Use the Execution Control steps to manipulate the execution of steps or devices.

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Sweep

Performs a sweep operation on the signals or parameters you select. A sweep iterates a set of measurement steps the number of times you specify. Each iteration modifies one or more parameters of one or more steps of the measurement.

Parameter	Description
Sweep Configuration	Contains the following options: • **Sweepable parameters**—Displays the parameters to sweep. The **Parameter Name** column displays the name of the parameter. The **Step Name** column displays the step that uses the parameter. The **Affected Output** column displays the output signals, if any, that the sweep operation affects. The **Alias** column displays the name to use in the **Formula** field when you reference the parameter. **Alias** displays a name only when you set **Type** to **Formula**. • **Add**—Adds a parameter to the **Sweepable parameters**. • **Remove**—Removes the selected parameter from the **Sweepable parameters**. • **Configuration**—Contains the following options: – **Type**—Specifies the type of sweep to perform. You can select from the following options: • **Linear**—(Default) Increments the selected parameter by a uniform value for each iteration of the sweep. The incremental value is the uniform distance between points that maps the **Number of points** between the **Start** and **Stop** values you specify. • **Exponential**—Increments the selected parameter by a value that grows exponentially with

each iteration of the sweep.

- **List of Points**—Adjusts the value of the selected parameter to the next value in the **Data Points** list with each iteration of the sweep.
- **From File**—Adjusts the value of the selected parameter to the next value in the file with each iteration of the sweep.
- **Formula**—Increments the selected parameter using the **Formula** you specify for the **Number of points** you specify.

– **Formula**—[Type: Formula] Specifies a formula you can use to control the selected parameter. Valid formula parameters are x, which follows a linear sweep characteristic based on the values specified by **Start x**, **Stop x** and **Number of Points**, and any alias x1, x2, and so on that is defined and visible in the **Alias** column. For example, if you sweep two parameters, x1 and x2, you can define a sweep formula to control the sweeping values of x2 depending on x and x1. For example, in the **Formula** field, enter 2*x1+3*x^2 to achieve a linear dependency to x1 and quadratic dependency to x. The **Sweep Points** graphs displays the resulting sweeping characteristic.

Note A formula can reference only an alias in a previously swept parameter.

– **Start**—[Type: Linear, Exponential] Specifies the start value of the selected

Sweepable parameter. The default is 1.

- **Stop**—[Type: Linear, Exponential] Specifies the stop value of the selected **Sweepable parameter**. The default is 2.
- **Start x**—[Type: Formula] Specifies the start value of the x parameter when you use the **Formula** sweep type. LabVIEW SignalExpress always sweeps the x parameter linearly.
- **Stop x**—[Type: Formula] Specifies the stop value of the x parameter when you use the **Formula** sweep type.
- **Number of Points**—[Type: Linear, Exponential, Formula] Specifies the number of points, including the start and stop point, to use to perform the sweep. The default is 2.
- **Sweep Points**—Displays a preview of the calculated points. The way LabVIEW SignalExpress calculates the points depends on the **Type** you specify.
    - [Type: Linear, Exponential]—Calculates points based on the values you specify in the **Start**, **Stop**, and **Number of Points** fields.
    - [Type: List of Points]—Calculates points based on the values in the **Data Points** list.
    - [Type: From File]—Calculates points based on the points specified in the file.
    - [Type: Formula]—Calculates points based on the values you specify in the **Start x**, **Stop x**, and **Number of Points** fields.
- **Data Points**—[Type: List of Points]

	Contains values that define the points in the sweep. – **Insert**—[Type: List of Points] Inserts a new point in the **Data Points** list above the selected point. – **Delete**—[Type: List of Points] Removes the selected point from the **Data Points** list. – **Path to sweep file**—[Type: From File] Specifies the location of the text file that contains the sweep points you want to use. The text file must contain a single column of values. You can add comments to each line of the file by using a semicolon delimiter between a sweep point and a comment.
Sweep Output	Contains the following options: • **Outputs of the sweep**—Displays the list of data to accumulate from each iteration of the sweep. The outputs you add to this list appear as outputs of the step in the Project View. • **Add**—Adds outputs to the **Outputs of the sweep** list. Clicking this button displays the Edit Sweep Output dialog box. • **Remove**—Removes the selected output from the **Outputs of the sweep** list. • **Edit**—Displays the Edit Sweep Output dialog box for the selected sweep output. • **Output options**—Contains the following options: – **Y-Axis Label (Range)**—Specifies a label for the y-axis of the selected output. – **X-Axis Label (Domain)**—Specifies a label for the x-axis of the selected output. – **Use default**—Specifies to whether to use the default label for the axis. Remove the checkmark from this

		checkbox to specify a custom label. – **Export iteration index**—Specifies whether to include the current iteration index as an output of the step.

Conditional Repeat

Repeats the sequence of steps inside the loop until one or more conditions are met. To add a condition, click the **Add** button in the **Input Configuration** section and select an appropriate input variable. The variable can be a Boolean or scalar result. You can use a Boolean signal as an exit condition directly, or you can compare a scalar value to a constant value. Select the comparison criteria using the **Criterion** pull-down menu.

You can combine several conditions into a single exit condition using **Group operation**. Select **AND** if you want to exit the loop when all the conditions are true. Select **OR** if you want to exit as soon as at least one of the conditions is met. You can invert the exit condition by removing the checkmark in the **Exit if True** checkbox.

Parameter	Description
Input Configuration	Contains the following options: • **Add**—Adds an input variable to the condition list. • **Remove**—Removes the input variable you select from the condition list. • **Loop conditions**—Displays the current conditions for the Conditional Repeat step. • **Input signal**—Specifies the input signal (variable) to be used with the specified conditions. • **Criterion**—Specifies the comparison criterion for the selected input scalar variable. • **Value**—Constant value used to compare with the selected variable. • **Invert input**—Inverts the selected input Boolean variable.
Exit Condition	Contains the following options: • **Group operation**—Combines several conditions into an Exit condition. Select **AND** if you want to exit the loop when all the conditions are true. Select **OR** if you want to exit as soon as at least one of the conditions is met.

	• **Exit if True**—Determines if the loop exits when the overall exit condition is true or false.

Trigger

Extracts a section of a continuous signal based on a specified trigger. This steps waits until a signal meets a trigger condition and returns a section of the signal that starts at the trigger point, or before the trigger point if you specify **Pre-trigger samples**. This step returns a triggered signal that always is the same size as the input signal before the trigger, so the input signal must provide enough data after the trigger occurs for the step to return a signal of that size. If the input signal does not provide enough data, this step times out.

Details

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Input Signal</td><td>Displays the input signal.
Displayed signal—Specifies the signal(s) to display in the preview graph(s). This option appears only when you select a group of signals for the input.
Note If the input signals include scalar values that depend on the values of input waveform signals, you cannot specify to display all signals in the preview graph(s).</td></tr>
<tr><td>Output Signal</td><td>Displays the output signal.</td></tr>
<tr><td>Input</td><td>Contains the following option:
Input signal—Specifies the input value.</td></tr>
<tr><td>Configuration</td><td>Contains the following options:
Trigger mode—Specifies which condition the specified signal must meet for LabVIEW SignalExpress to begin extracting the signal.
Positive Edge—Specifies for the trigger to occur when a signal crosses the specified Level with a positive slope.
Negative Edge—Specifies for the trigger to occur when a signal crosses the specified Level with a negative slope.
Entering Window—Specifies for the</td></tr>
</table>

trigger to occur when a signal enters the window between the specified **High level** and **Low level** values.

   - **Leaving Window**—Specifies for the trigger to occur when a signal exits the window between the specified **High level** and **Low level** values.

- **Level**—Specifies the level that the signal must meet to trigger the input signal.
- **High level**—Specifies the higher limit of the range to trigger the specified signal.
- **Low level**—Specifies the lower limit of the range to trigger the specified signal.
- **Hysteresis**—Specifies the amount above and below **Trigger level** through which the specified signal must pass before a trigger level crossing is detected. The default is 0. Use trigger hysteresis to prevent noise from causing a false trigger.
- **Trigger signal**—Specifies the channel to use to search for the trigger when the input is a grouped signal. The default is the first channel in the group.
- **Pre-trigger samples**—Specifies the number of pre-trigger samples per channel to acquire before the reference trigger point. Devices that NI-DAQmx supports require **Pre-trigger samples** to have a minimum value of 2. Post-trigger samples equal **Samples to read** – **Pre-trigger samples**. The default is 2.
- **Timeout (s)**—Specifies the hardware timeout value in seconds. The default is 10.
- **Trigger found**—Returns TRUE if a trigger has been found in the input signal. Returns FALSE if no trigger is found.

Trigger Details

This step can return a timeout error under the following conditions:

- You specify a trigger **Level** that the input signal does not meet.
- A trigger occurs but the input signal does not provide enough data after the trigger occurs for the step to return a signal of the same size as the input signal.

Sequence

Controls the execution of a project by alternately executing and pausing steps. The Sequence step can pause the execution of steps in a project without stopping the execution of the entire project. For example, you can use the Sequence step to pause the execution of a step that acquires a signal from a hardware device so that another step in the project can use the same hardware device.

If you select **Run preceding steps before following steps**, the Sequence step allows steps that precede the Sequence step to execute once, then pauses the execution of those steps and allows the steps that follow the Sequence step to execute once, and so on.

If you select **Start a sequence** or **End a sequence**, the Sequence step becomes the beginning or end of a sequence and does not affect the execution of any steps that appear before or after the sequence in the Project View. Use the **Start a sequence** and **End a sequence** options when you want to allow steps outside of the sequence to run continuously, such as if you want to generate a signal continuously.

Note When you use the Sequence step to reuse hardware, you cannot perform a continuous signal acquisition because LabVIEW SignalExpress stops and starts the hardware device.

Parameter	Description
Configuration	Contains the following options: • **Run preceding steps before following steps**—Specifies whether Sequence forces all the steps that precede it in the sequence to run before all the steps that follow it. – **Allow hardware reuse**—Specifies whether steps that follow the Sequence step in the current sequence can use the same hardware as steps that precede the Sequence step. Place a checkmark in this checkbox to enable hardware reuse. • **Start a sequence**—Specifies whether the Sequence step starts a new sequence. If you

	select this option, the Sequence step starts a new sequence in a new execution loop. • **End a sequence**—Specifies whether the Sequence step ends a sequence. If you select this option, the Sequence step ends the current sequence and closes the current execution loop. • **Update signal views after each iteration**—Updates the signal views, including displays on the Data View tab and preview graphs on the Step Setup tab, each time the Sequence step executes. If you select **Display message at each iteration**, this option enables the user to see updated signal values when LabVIEW SignalExpress pauses the project.
Action	Contains the following options: • **Display message at each iteration**—Specifies whether LabVIEW SignalExpress pauses the execution of the sequence and displays a message to the user each time the Sequence step executes. If you place a checkmark in this checkbox, the LabVIEW SignalExpress dialog box appears at each iteration to display the message and prompt the user to continue or stop running the project. This dialog box also gives the user the option to create a snapshot of the signals in the project. – **Message Text**—Specifies the text to display in the message that appears each time the Sequence step executes.
Timing	Contains the following options: • **Pre-execution delay**—Specifies the amount of time in milliseconds to wait before the step executes. • **Post-execution delay**—Specifies the amount of time in milliseconds to wait after the step executes.

Run LabVIEW VI

Use the Run LabVIEW VI steps to run a LabVIEW VI in the LabVIEW SignalExpress environment.

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW SignalExpress Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Run LabVIEW 7.1 VI

Runs a LabVIEW 7.1 VI in the LabVIEW SignalExpress environment. LabVIEW SignalExpress assumes all controls on the connector pane of the VI you specify are parameters. If there are inputs available, use the **Connect Input** button to specify a control LabVIEW SignalExpress can use to pass data to your VI.

You must use the version of the Run LabVIEW VI step that matches the version of LabVIEW you saved your VI in. For example, if you saved a VI in LabVIEW 7.1, you must use the Run LabVIEW 7.1 VI step.

If you use a VI that contains a parameter with an X-Y array data type, the output must be a cluster indicator, not a graph indicator. The x-array must be the first component in the cluster, and the y-array must be the second component in the cluster. You must represent both components in the cluster as numeric doubles. LabVIEW SignalExpress can accept a variety of data types.

You can build a Run LabVIEW VI step from an existing VI or you can start from a template.

Parameter	Description
Settings	Contains the following options: • **Select VI**—Specifies the path to the VI you want to use as a step in the LabVIEW SignalExpress environment. • **Run this step automatically**—Runs the VI as soon as you make a change to the front panel. • **Refresh VI**—Reloads the VI from disk, scans the VI for any changes made while using LabVIEW SignalExpress, and updates the Run LabVIEW VI step with any changes. If you are running a project in LabVIEW SignalExpress, you must click the **Stop** button before you click the **Refresh VI** button for the selected VI to successfully update. • **Control name**—Specifies which inputs of the VI to designate as signal inputs from the LabVIEW SignalExpress project. If you want to pass data from a different step of the project into the VI, use an

<table>
<tr><td></td><td>input signal. LabVIEW SignalExpress assumes all controls on the connector pane of the VI you specify are parameters. An input signal is a value that comes from a previous step. You cannot edit that input signal on the front panel of the VI.
Connect Input—Specifies a control LabVIEW SignalExpress can use to pass data to the VI.
Disconnect Input—Removes the input from the Control Name listbox.
Input signal—Lists a reference to a signal from a previous step.
Outputs—Lists the recognized outputs of the VI that return data to the project. These outputs must be on the front panel of the VI, match one of the LabVIEW SignalExpress signal types, and be connected to the VI connector pane.
Strict type of output data—Specifies the signal type of the output from the VI. You must specify a strict signal type to guarantee that other steps in the project can accept the signal as an input and properly operate on it.</td></tr>
<tr><td>Configure VI</td><td>Displays the front panel of the VI you selected on the Settings tab.</td></tr>
<tr><td>Execution Control</td><td>Contains the following options:
Start this step after—Makes the step wait until another step has started before executing. You can make the step wait on any other hardware step in the project by selecting the step to wait on from the pull-down menu.

You can use this option to force an acquisition device to start after a generation device starts. You also can use this option to ensure that a device generating a trigger signal starts after the device receiving the signal, which avoids sending the signal before the receiver is ready.
Step to wait for—Lists the possible steps for which</td></tr>
</table>

this step can wait.

- **Pre-execution delay (ms)**—Specifies the amount of time to wait before the step executes. If you configure the step to start after another step, the delay represents the amount of time to wait after the specified step starts.
- **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes.
- **Timing type**—Contains the following options that specify how the output signal of the step is timed:
  - **Untimed**—Specifies that the output signal is not timed. Select this option if the output signal is not a scalar value or a time-continuous waveform.
  - **Continuous**—Specifies that the output signal is timed continuously. Select this option if the output signal is a time-continuous waveform.
  - **Periodic**—Specifies that the output signal is timed periodically. Select this option if the output signal is a scalar value and you want to force the step to run at a rate you specify in **Sample period (s)**.
  - **Sample period (s)**—[Timing type: Periodic] Specifies the sample period to use for a periodic output signal. Set **Sample period (s)** to 0 to run the step as fast as possible.

Run LabVIEW 8.0 VI

Runs a LabVIEW 8.0 VI in the LabVIEW SignalExpress environment. LabVIEW SignalExpress assumes all controls on the connector pane of the VI you specify are parameters. If there are inputs available, use the **Connect Input** button to specify a control LabVIEW SignalExpress can use to pass data to your VI.

You must use the version of the Run LabVIEW VI step that matches the version of LabVIEW you saved your VI in. For example, if you saved a VI in LabVIEW 7.1, you must use the Run LabVIEW 7.1 VI step.

If you use a VI that contains a parameter with an X-Y array data type, the output must be a cluster indicator, not a graph indicator. The x-array must be the first component in the cluster and the y-array must be the second component in the cluster. You must represent both components in the cluster as numeric doubles. LabVIEW SignalExpress can accept a variety of data types.

You can build a Run LabVIEW VI step from an existing VI or you can start from a template.

Parameter	Description
Settings	Contains the following options: • **Select VI**—Specifies the path to the VI you want to use as a step in the LabVIEW SignalExpress environment. • **Run this step automatically**—Runs the VI as soon as you make a change to the front panel. • **Refresh VI**—Reloads the VI from disk, scans the VI for any changes made while using LabVIEW SignalExpress, and updates the Run LabVIEW VI step with any changes. If you are running a project in LabVIEW SignalExpress, you must click the **Stop** button before you click the **Refresh VI** button for the selected VI to successfully update. • **Control name**—Specifies which inputs of the VI to designate as signal inputs from the LabVIEW SignalExpress project. If you want to pass data from a different step of the project into the VI, use an

	input signal. LabVIEW SignalExpress assumes all controls on the connector pane of the VI you specify are parameters. An input signal is a value that comes from a previous step. You cannot edit that input signal on the front panel of the VI. • **Connect Input**—Specifies a control LabVIEW SignalExpress can use to pass data to the VI. • **Disconnect Input**—Removes the input from the **Control Name** listbox. • **Input signal**—Lists a reference to a signal from a previous step. • **Outputs**—Lists the recognized outputs of the VI that return data to the project. These outputs must be on the front panel of the VI, match one of the LabVIEW SignalExpress signal types, and be connected to the VI connector pane. • **Strict type of output data**—Specifies the signal type of the output from the VI. You must specify a strict signal type to guarantee that other steps in the project can accept the signal as an input and properly operate on it.
Configure VI	Displays the front panel of the VI you selected on the **Settings** tab.
Execution Control	Contains the following options: • **Start this step after**—Makes the step wait until another step has started before executing. You can make the step wait on any other hardware step in the project by selecting the step to wait on from the pull-down menu. You can use this option to force an acquisition device to start after a generation device starts. You also can use this option to ensure that a device generating a trigger signal starts after the device receiving the signal, which avoids sending the signal before the receiver is ready. • **Step to wait for**—Lists the possible steps for which

		this step can wait. • **Pre-execution delay (ms)**—Specifies the amount of time to wait before the step executes. If you configure the step to start after another step, the delay represents the amount of time to wait after the specified step starts. • **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes. • **Timing type**—Contains the following options that specify how the output signal of the step is timed: – **Untimed**—Specifies that the output signal is not timed. Select this option if the output signal is not a scalar value or a time-continuous waveform. – **Continuous**—Specifies that the output signal is timed continuously. Select this option if the output signal is a time-continuous waveform. – **Periodic**—Specifies that the output signal is timed periodically. Select this option if the output signal is a scalar value and you want to force the step to run at a rate you specify in **Sample period (s)**. – **Sample period (s)**—[Timing type: Periodic] Specifies the sample period to use for a periodic output signal. Set **Sample period (s)** to 0 to run the step as fast as possible.

Run LabVIEW 8.2 VI

Runs a LabVIEW 8.2 VI in the LabVIEW SignalExpress environment. LabVIEW SignalExpress assumes all controls on the connector pane of the VI you specify are parameters. If there are inputs available, use the **Connect Input** button to specify a control LabVIEW SignalExpress can use to pass data to your VI.

You must use the version of the Run LabVIEW VI step that matches the version of LabVIEW you saved your VI in. For example, if you saved a VI in LabVIEW 7.1, you must use the Run LabVIEW 7.1 VI step.

If you use a VI that contains a parameter with an X-Y array data type, the output must be a cluster indicator, not a graph indicator. The x-array must be the first component in the cluster and the y-array must be the second component in the cluster. You must represent both components in the cluster as numeric doubles. LabVIEW SignalExpress can accept a variety of data types.

You can build a Run LabVIEW VI step from an existing VI or you can start from a template.

Parameter	Description
Settings	Contains the following options: • **Select VI**—Specifies the path to the VI you want to use as a step in the LabVIEW SignalExpress environment. • **Run this step automatically**—Runs the VI as soon as you make a change to the front panel. • **Refresh VI**—Reloads the VI from disk, scans the VI for any changes made while using LabVIEW SignalExpress, and updates the Run LabVIEW VI step with any changes. If you are running a project in LabVIEW SignalExpress, you must click the **Stop** button before you click the **Refresh VI** button for the selected VI to successfully update. • **Control name**—Specifies which inputs of the VI to designate as signal inputs from the LabVIEW SignalExpress project. If you want to pass data from a different step of the project into the VI, use an

	input signal. LabVIEW SignalExpress assumes all controls on the connector pane of the VI you specify are parameters. An input signal is a value that comes from a previous step. You cannot edit that input signal on the front panel of the VI. • **Connect Input**—Specifies a control LabVIEW SignalExpress can use to pass data to the VI. • **Disconnect Input**—Removes the input from the **Control Name** listbox. • **Input signal**—Lists a reference to a signal from a previous step. • **Outputs**—Lists the recognized outputs of the VI that return data to the project. These outputs must be on the front panel of the VI, match one of the LabVIEW SignalExpress signal types, and be connected to the VI connector pane. • **Strict type of output data**—Specifies the signal type of the output from the VI. You must specify a strict signal type to guarantee that other steps in the project can accept the signal as an input and properly operate on it.
Configure VI	Displays the front panel of the VI you selected on the **Settings** tab.
Execution Control	Contains the following options: • **Start this step after**—Makes the step wait until another step has started before executing. You can make the step wait on any other hardware step in the project by selecting the step to wait on from the pull-down menu. You can use this option to force an acquisition device to start after a generation device starts. You also can use this option to ensure that a device generating a trigger signal starts after the device receiving the signal, which avoids sending the signal before the receiver is ready. • **Step to wait for**—Lists the possible steps for which

		this step can wait. • **Pre-execution delay (ms)**—Specifies the amount of time to wait before the step executes. If you configure the step to start after another step, the delay represents the amount of time to wait after the specified step starts. • **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes. • **Timing type**—Contains the following options that specify how the output signal of the step is timed: – **Untimed**—Specifies that the output signal is not timed. Select this option if the output signal is not a scalar value or a time-continuous waveform. – **Continuous**—Specifies that the output signal is timed continuously. Select this option if the output signal is a time-continuous waveform. – **Periodic**—Specifies that the output signal is timed periodically. Select this option if the output signal is a scalar value and you want to force the step to run at a rate you specify in **Sample period (s)**. – **Sample period (s)**—[Timing type: Periodic] Specifies the sample period to use for a periodic output signal. Set **Sample period (s)** to 0 to run the step as fast as possible.

Run LabVIEW 8.5 VI

Runs a LabVIEW 8.5 VI in the LabVIEW SignalExpress environment. LabVIEW SignalExpress assumes all controls on the connector pane of the VI you specify are parameters. If there are inputs available, use the **Connect Input** button to specify a control LabVIEW SignalExpress can use to pass data to your VI.

You must use the version of the Run LabVIEW VI step that matches the version of LabVIEW you saved your VI in. For example, if you saved a VI in LabVIEW 7.1, you must use the Run LabVIEW 7.1 VI step.

If you use a VI that contains a parameter with an X-Y array data type, the output must be a cluster indicator, not a graph indicator. The x-array must be the first component in the cluster and the y-array must be the second component in the cluster. You must represent both components in the cluster as numeric doubles. LabVIEW SignalExpress can accept a variety of data types.

You can build a Run LabVIEW VI step from an existing VI or you can start from a template.

Parameter	Description
Settings	Contains the following options: • **Select VI**—Specifies the path to the VI you want to use as a step in the LabVIEW SignalExpress environment. • **Run this step automatically**—Runs the VI as soon as you make a change to the front panel. • **Refresh VI**—Reloads the VI from disk, scans the VI for any changes made while using LabVIEW SignalExpress, and updates the Run LabVIEW VI step with any changes. If you are running a project in LabVIEW SignalExpress, you must click the **Stop** button before you click the **Refresh VI** button for the selected VI to successfully update. • **Control name**—Specifies which inputs of the VI to designate as signal inputs from the LabVIEW SignalExpress project. If you want to pass data from a different step of the project into the VI, use an

	input signal. LabVIEW SignalExpress assumes all controls on the connector pane of the VI you specify are parameters. An input signal is a value that comes from a previous step. You cannot edit that input signal on the front panel of the VI. • **Connect Input**—Specifies a control LabVIEW SignalExpress can use to pass data to the VI. • **Disconnect Input**—Removes the input from the **Control Name** listbox. • **Input signal**—Lists a reference to a signal from a previous step. • **Outputs**—Lists the recognized outputs of the VI that return data to the project. These outputs must be on the front panel of the VI, match one of the LabVIEW SignalExpress signal types, and be connected to the VI connector pane. • **Strict type of output data**—Specifies the signal type of the output from the VI. You must specify a strict signal type to guarantee that other steps in the project can accept the signal as an input and properly operate on it.
Configure VI	Displays the front panel of the VI you selected on the **Settings** tab.
Execution Control	Contains the following options: • **Start this step after**—Makes the step wait until another step has started before executing. You can make the step wait on any other hardware step in the project by selecting the step to wait on from the pull-down menu. You can use this option to force an acquisition device to start after a generation device starts. You also can use this option to ensure that a device generating a trigger signal starts after the device receiving the signal, which avoids sending the signal before the receiver is ready. • **Step to wait for**—Lists the possible steps for which

		this step can wait. • **Pre-execution delay (ms)**—Specifies the amount of time to wait before the step executes. If you configure the step to start after another step, the delay represents the amount of time to wait after the specified step starts. • **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes. • **Timing type**—Contains the following options that specify how the output signal of the step is timed: – **Untimed**—Specifies that the output signal is not timed. Select this option if the output signal is not a scalar value or a time-continuous waveform. – **Continuous**—Specifies that the output signal is timed continuously. Select this option if the output signal is a time-continuous waveform. – **Periodic**—Specifies that the output signal is timed periodically. Select this option if the output signal is a scalar value and you want to force the step to run at a rate you specify in **Sample period (s)**. – **Sample period (s)**—[Timing type: Periodic] Specifies the sample period to use for a periodic output signal. Set **Sample period (s)** to 0 to run the step as fast as possible.

Run LabVIEW 8.6 VI

Runs a LabVIEW 8.6 VI in the LabVIEW SignalExpress environment. LabVIEW SignalExpress assumes all controls on the connector pane of the VI you specify are parameters. If there are inputs available, use the **Connect Input** button to specify a control LabVIEW SignalExpress can use to pass data to your VI.

You must use the version of the Run LabVIEW VI step that matches the version of LabVIEW you saved your VI in. For example, if you saved a VI in LabVIEW 7.1, you must use the Run LabVIEW 7.1 VI step.

If you use a VI that contains a parameter with an X-Y array data type, the output must be a cluster indicator, not a graph indicator. The x-array must be the first component in the cluster and the y-array must be the second component in the cluster. You must represent both components in the cluster as numeric doubles. LabVIEW SignalExpress can accept a variety of data types.

You can build a Run LabVIEW VI step from an existing VI or you can start from a template.

Parameter	Description
Settings	Contains the following options: • **Select VI**—Specifies the path to the VI you want to use as a step in the LabVIEW SignalExpress environment. • **Run this step automatically**—Runs the VI as soon as you make a change to the front panel. • **Refresh VI**—Reloads the VI from disk, scans the VI for any changes made while using LabVIEW SignalExpress, and updates the Run LabVIEW VI step with any changes. If you are running a project in LabVIEW SignalExpress, you must click the **Stop** button before you click the **Refresh VI** button for the selected VI to successfully update. • **Control name**—Specifies which inputs of the VI to designate as signal inputs from the LabVIEW SignalExpress project. If you want to pass data from a different step of the project into the VI, use an

	input signal. LabVIEW SignalExpress assumes all controls on the connector pane of the VI you specify are parameters. An input signal is a value that comes from a previous step. You cannot edit that input signal on the front panel of the VI. • **Connect Input**—Specifies a control LabVIEW SignalExpress can use to pass data to the VI. • **Disconnect Input**—Removes the input from the **Control Name** listbox. • **Input signal**—Lists a reference to a signal from a previous step. • **Outputs**—Lists the recognized outputs of the VI that return data to the project. These outputs must be on the front panel of the VI, match one of the LabVIEW SignalExpress signal types, and be connected to the VI connector pane. • **Strict type of output data**—Specifies the signal type of the output from the VI. You must specify a strict signal type to guarantee that other steps in the project can accept the signal as an input and properly operate on it.
Configure VI	Displays the front panel of the VI you selected on the **Settings** tab.
Execution Control	Contains the following options: • **Start this step after**—Makes the step wait until another step has started before executing. You can make the step wait on any other hardware step in the project by selecting the step to wait on from the pull-down menu. You can use this option to force an acquisition device to start after a generation device starts. You also can use this option to ensure that a device generating a trigger signal starts after the device receiving the signal, which avoids sending the signal before the receiver is ready. • **Step to wait for**—Lists the possible steps for which

this step can wait.

- **Pre-execution delay (ms)**—Specifies the amount of time to wait before the step executes. If you configure the step to start after another step, the delay represents the amount of time to wait after the specified step starts.
- **Post-execution delay (ms)**—Specifies the amount of time to wait after the step executes.
- **Timing type**—Contains the following options that specify how the output signal of the step is timed:
  - **Untimed**—Specifies that the output signal is not timed. Select this option if the output signal is not a scalar value or a time-continuous waveform.
  - **Continuous**—Specifies that the output signal is timed continuously. Select this option if the output signal is a time-continuous waveform.
  - **Periodic**—Specifies that the output signal is timed periodically. Select this option if the output signal is a scalar value and you want to force the step to run at a rate you specify in **Sample period (s)**.
  - **Sample period (s)**—[Timing type: Periodic] Specifies the sample period to use for a periodic output signal. Set **Sample period (s)** to 0 to run the step as fast as possible.

LabVIEW SignalExpress Environment

The projects you create in the LabVIEW SignalExpress environment include the steps you select from the Add Step menu. You display the results of the measurements in graphs and tables.

You can edit projects while the project runs, and the results update.

If you installed National Instruments LabVIEW, you can convert projects into VIs or use the Run LabVIEW VI step to import VIs for use in LabVIEW SignalExpress projects.

Note You must use LabVIEW 7.1 or later to create a Run LabVIEW VI step. You cannot run a VI that was saved for an earlier version of LabVIEW using the Run LabVIEW VI step. You must save your VI in the actual version of LabVIEW you want to run.

Channel View

The Channel View is a central location for viewing and configuring hardware and shared variables. When you launch LabVIEW SignalExpress, the application automatically detects installed or simulated NI-DAQmx devices, NI-DMM devices, and NI switch modules and displays the devices in the Channel View. You also can import and export Channel View data from Microsoft Excel spreadsheets.

The Channel View does not display analog output (AO) modules.

In the default layout, the Channel View is a supplementary view that appears below the primary view as a table with two columns: **Physical Channel** and **Acquire**. LabVIEW SignalExpress displays the devices it detects in the **Physical Channel** column.

Note If LabVIEW SignalExpress does not detect any installed or simulated NI-DAQmx devices, NI-DMM devices, or NI switch modules, the Channel View does not appear in the default layout. If you display the Channel View with no supported hardware present, the Channel View appears empty.

Use the **View** pull-down menu at the top of the Channel View to specify whether to display hardware or shared variables. If you select **Shared Variables** from the **View** pull-down menu, the **Shared Variable Name** and the **Sample Period (s)** columns appear.

The **Acquire** column contains checkboxes that specify whether to acquire signals from hardware or read the values of shared variables. When you place a checkmark in the **Acquire** checkbox for a hardware item, LabVIEW SignalExpress adds an Acquire Signals step to the Project View and additional columns appear with item-specific configuration options. When you place a checkmark in the **Acquire** checkbox for a shared variable, LabVIEW SignalExpress adds a Read Shared Variables step to the Project View or, if a Read Shared Variables Step already exists, adds the value of the shared variable as an output of the existing step.

If the Channel View is not visible, select **View»Channel View** to display the Channel View.

Configuring Items from the Properties Window

You can use the Properties window to configure channels, devices, or shared variables. The **Properties** window displays the configuration options for the device(s), channel(s), or shared variable(s) you select in the Channel View. If you select multiple items, the **Properties** window displays all the configuration options the items share so you quickly can update values for all the items. For example, if the Channel View shows that you are measuring voltage on 10 channels and you want to measure resistance on those 10 channels, you can select all 10 channels and use the **Properties** window to update the measurement type once instead of on each individual channel.

If the **Properties** window is not visible, select **View»Properties** to display the **Properties** window. For hardware devices, as you make changes in the **Properties** window, the Channel View table displays new columns with the appropriate configurable options for the selected measurement.

Logged Data Window

The **Logged Data** window appears below the Project View in LabVIEW SignalExpress and displays a list of all logged data and snapshots from the current project sorted by the time at which you recorded the log or took the snapshot. You can use the **Logged Data** window to view, export, and manage logged data and snapshots.

When you create a new data log or snapshot, the name of the new data log or snapshot automatically appears in the **Logged Data** window. Expand the name to display the signals that the data log or snapshot contains.

Managing Logged Data

Each data log has a top-level, user-defined name. The default name is the timestamp, but you can right-click the timestamp and select **Rename** from the shortcut menu to rename a log.

Note You also can use the Recording Options tab to name a log before you record the log.

The name of the active log appears in bold in the **Logged Data** window. Right-click the name of a data log and select **Make Active Log** to make that log the active log. If you are viewing a log on the Data View tab, the display updates to display the active log.

When you expand a data log in the **Logged Data** window, you can right-click a signal within the data log and select from the following shortcut menu options to manage the logged signal:

- **Properties**—Displays a summary of the properties of the logged signal.
- **Show Alarms and Events**—Displays a list of all the alarms and events in the logged signal.
- **Open Folder**—Navigates to the location of the logged signal on disk.
- **Convert to ASCII**—Converts the logged signal to an ASCII file.
- **Export to Microsoft Excel**—Exports the logged data to Microsoft Excel. This option exports every sample in the log.

Note When you export data to Microsoft Excel, LabVIEW SignalExpress copies data samples to a clipboard for export. Use the **Maximum Clipboard Data Export Size** option on the Data page of the Options dialog box to increase the number of data samples the clipboard can contain.

- **Open in DIAdem**—Opens the logged signal in DIAdem.
- **Make Log Viewable**—Specifies for LabVIEW SignalExpress to process the log so you can view the log on a reasonable scale in a display on the **Data View** tab. Select this option if you set **Prepare log data for viewing** to **Never** on the Logging page of the Options dialog box.

Managing Snapshots

Each snapshot has a top-level, user-defined name. The default name is the timestamp, but you can right-click the timestamp and select **Rename** from the shortcut menu to rename a snapshot.

Note You also can use the Create Snapshot dialog box to name a snapshot before you create the snapshot.

When you expand a snapshot in the **Logged Data** window, you can right-click a signal within the snapshot and select from the following shortcut menu options to manage the signal:

- **Send To**—Sends the signal to an analysis step.
- **Probe**—Displays a Data Probe window with detailed information about the signal.
- **Copy Value**—Copies the signal value as text that you can paste into a text file.
- **Save Value**—Saves the snapshot as a text file.
- **Delete**—Deletes the signal from the snapshot.
- **Rename**—Renames the signal.

Operator Interface

Use the Operator Interface view to create an operator interface containing controls that you can use to modify step parameters. When you run a project in operator mode, the only values you can change are values of step parameters that you bind to controls on the operator interface.

Select **View»Operator Interface** to display the Operator Interface view.

The Toolbox window contains the controls you can add to the operator interface. Drag a control from the **Toolbox** window to the Operator Interface view to add the control to the operator interface. You can click the arrow icon (▸) that appears on an operator interface control to configure basic properties of the control, or you can use the Properties window to view and configure all the properties of an operator interface control.

Note You cannot edit the controls on an operator interface when a project is in operator mode.

You cannot add indicators to the Operator Interface view. Use displays on the Data View tab to see how changing the value of an operator interface control effects signals in the project.

Configuring the Project View

You can use the **Options** dialog box to configure how the steps in the Project View appear. Select **Tools»Options** to display the **Options** dialog box. On the **General** page, navigate to the **Project View** section.

The **Show input and output signals for all steps** option displays all signals for all steps or all signals for the selected step.

The **Show large icons in the project view** option displays all steps with large icons or the selected step with a large icon.

Copying and Pasting Steps

When you copy and insert a step, LabVIEW SignalExpress updates the step and subsequent steps that inherit from the copied step to process the correct signals.

Complete the following steps to copy and insert a step.

1. In the Project View, right-click the step you want to copy and select **Copy** from the shortcut menu.
2. Right-click a step in the Project View and select **Paste Before Selected Step** or **Paste After Selected Step** from the shortcut menu to insert the copied step before or after the step you selected.

Signal Types in LabVIEW SignalExpress

LabVIEW SignalExpress categorizes signals based on their type, such as time-domain signals, frequency-domain signals, scalar values, or Boolean values. Time-domain signals appear on a graph where a signal value (such as amplitude) corresponds to a time. Frequency-domain signals appear on a graph where the level of a signal corresponds to a frequency value. Scalar values appear in tables as single values or a collection of single values, such as the result of a DC measurement or a tone-frequency measurement. Boolean values appear as vertical or horizontal LEDs. LabVIEW SignalExpress can manage the different signal types in the following ways:

- You can process only signals that make sense for a specific step. For example, you only can perform a power spectrum operation on a time-domain signal.
- You cannot mix signal types that are incompatible. For example, you cannot add time-domain signal values to the resulting values of a power spectrum.
- You can add a phase signal represented in degrees to a phase signal represented in radians. LabVIEW SignalExpress converts the different signal types.
- You cannot display incompatible signals on the same graph. For example, you cannot display a time signal and the result of a power spectrum on the same graph.

Steps Can Process Different Signal Types

Some steps can process more than one signal type. For example, the Scaling and Conversion step can scale a time-domain signal (applying gain and offset values) or can scale a frequency-domain signal.

The available scaling operations depend on the type and unit of the signal. If the input signal is a frequency-domain magnitude signal, you can apply a gain to the magnitude signal. If the input signal is a frequency-domain phase signal, you can apply a delay correction value.

LabVIEW SignalExpress Data Types

The following table lists the most common data types LabVIEW SignalExpress uses.

Graphic	Description
	Time Waveform, Real
	Time Waveform, Real, 1D Array
	Time-XY Waveform, Real
	Time Digital Waveform,
	Frequency Spectrum, Magnitude
	Frequency Spectrum, Phase
	Frequency Spectrum, Coherence
	Frequency-XY Waveform, Power Linear
	Frequency-XY Waveform, Power (dB)
	Frequency-XY Waveform, Magnitude
	Frequency-XY Waveform, Magnitude (dB)
	Frequency-XY Waveform, Phase (rad.)
	Frequency-XY Waveform, Phase (deg.)
	Frequency Waveform, Ratio
	Histogram, Generic
	Histogram, Generic (%)
	Scalar, U32
	Scalar, Double
	Scalar, Double, Array 1D
	Boolean
	String

Properties Window

The **Properties** window displays editable attributes of devices, channels, or operator interface controls. You can use the **Properties** window to configure multiple devices or channels that you select in the Channel View or to configure controls you add to the Operator Interface view.

If you select an item in either the Channel View or the Operator Interface view, the **Properties** window displays the name and attributes of the item you select. If you select multiple items, the **Properties** window displays the attributes that are common to all the selected items. You cannot change the values of attributes that appear disabled, or grayed out.

Select **View»Properties** to display the **Properties** window.

Configuring Device Channels Using the Properties Window

You can use the Properties window to configure devices and device channels when you acquire signals using the Channel View. Use the **Properties** window to quickly configure multiple channels when you perform multi-channel signal acquisitions. Complete the following steps to configure multiple device channels simultaneously.

1. If the Channel View is not visible, select **View»Channel View** to display the Channel View.
2. Select **View»Properties** or click the **Properties** tab that appears in the bottom left corner of the application window to display the **Properties** window.
3. In the Channel View, click the expand symbol for the device from which you want to acquire signals to display the channels of the device.
4. Select the channels from which you want to acquire signals. Hold down the <Ctrl> key as you select the channels you want to configure to select multiple channels. You also can select a channel, hold down the <Shift> key, and select another channel to select all the channels between the two you click, inclusive.

   The attributes you can configure for the selected channels appear enabled in the **Properties** window. Settings that you cannot configure for all selected channels appear grayed out. Grayed out attributes usually are attributes that you configured at the device level and that you cannot change for individual channels.
5. In the **Properties** window, set the enabled settings to the values you want to use. LabVIEW SignalExpress updates all the selected channels with the values you specify.

Toolbox Window

The **Toolbox** window contains a list of controls you can add to an operator interface. Controls include knobs, slides, switches, rings, text controls, and labels. Operator interface controls resemble the controls that appear on the front panel of an instrument. An operator interface allows a user to change step parameters when a project is in operator mode. Use the **Toolbox** window with the Operator Interface view to create an operator interface. Select a control in the **Toolbox** window and drag the control to the Operator Interface view to add the control to the operator interface.

Select **View»Toolbox** to display the **Toolbox** window.

Data View

The **Data View** tab displays data in LabVIEW SignalExpress. You can drag a signal directly from a step in the Project View to the **Data View** tab to display the value of the signal. The **Data View** tab can display signals in several formats, including graphs, charts, and various numeric representations.

When you drag a signal to the **Data View** tab, the signal appears in a new or existing display depending on whether a display showing that type of data already exists. By default, the **Data View** tab appears with a graph display, but LabVIEW SignalExpress automatically updates the type of the display to fit the data type of the signal you add.

A project can contain multiple **Data View** tabs, and one **Data View** tab can contain multiple displays. Right-click a display and select **Data View»New Data View** from the shortcut menu to create a new **Data View** tab. To remove a **Data View** tab, either close the tab or right-click a display on the tab and select **Data View»Remove Data View** from the shortcut menu. You also can use the Data View menu to add, remove, or modify the appearance of the **Data View** tab.

If the **Data View** tab is not visible, select **View»Data View** to display the tab.

Viewing Data in a Playback Work Area

In a Playback work area, the **Data View** tab appears with a playback toolbar you can use to navigate logged data. Drag a log from the Logged Data window to a display on the **Data View** tab to view the logged data. You then can use the buttons on the playback toolbar to play back the log, play back the log repeatedly, or update the display to show the next or previous iteration of the log. You can use the down arrow to set the speed of playback. The playback toolbar also includes a time line you can use to scroll to a specific point in a log.

Event Viewer

When you display logged signals on graph displays on the Data View tab, you can use the Event Viewer to display events such as errors, warnings, and alarms that occurred during the log. Right-click the graph of a logged signal and select **Visible Items»Event Viewer** to display the Event Viewer. For each event, the Event Viewer displays the type of the event, the time the event occurred, and a brief description of the event. Double-click an event in the Event Viewer to zoom in on the signal where the event occurred.

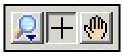
Graph Legend

Right-click a graph in the Data View and select **Visible Items»Legend** to open the graph legend. The graph legend lists every signal displayed in the graph and the corresponding plot color. To remove a signal from the graph, remove the checkmark from the checkbox next to the signal name. Click the plot color next to a signal to change the color of the signal.

The order in which the graph legend lists signals is the plot order of the graph. You can use the Signal Order page of the Display Properties dialog box to change the plot order of a graph.

Graph Palette

The graph palette appears below and to the left of a graph on a graph or chart display or on the Project Documentation tab. Use the graph palette, shown as follows, to specify how you interact with a graph.



With the graph palette, you can move cursors, zoom, and pan the display. The graph palette appears with the following buttons, in order from left to right:

- **Zoom**—Zooms in and out of the display. Use the pull-down menu that appears when you click this button to select a zoom method.
- **Cursor Movement Tool**—Moves cursors on the display.
- **Panning Tool**—Picks up the plot and moves it around on the display.

Click a button on the graph palette to enable zooming, moving cursors, or panning.

Graph Timestamp

Time-domain graphs in LabVIEW SignalExpress have three timestamp options for displaying signals. LabVIEW SignalExpress displays graphs of time-domain signals with timestamps that provide the clearest view of the signal or signals by default. You can change the timestamp of a graph by right-clicking the graph, selecting **Time Stamp** from the shortcut menu, and selecting one of the following timestamp options:

- **Ignore**—All the plots on the graph start at zero regardless of the actual timestamp of the plotted signals. Select this option to display multiple plots with unrelated timestamps, such as displaying an acquisition result and a simulation signal loaded from a file.
- **Absolute**—All the plots on the graph display signals that show the actual timestamp of each waveform. This is the default setting for graphs of logged data. Use this option if you want to display a waveform acquired using a triggered digitizer where the timestamp reference (zero) is linked to the trigger point.
- **Relative**—All plots on the graph display signals relative to the start time of the reference signal you select. The reference signal starts at zero. For example, if you plot two waveforms time-stamped at 15 and 13 and select the first waveform as the reference, the two plots start at 0 and –2. This is the default setting for graphs of only one signal. Use this option if you want to display multiple channels that a multiplexed MIO board acquires and display the timing relationship between the different channels to correct for the interchannel delays.

Preview Graph

The Preview Graph provides a method for panning through and zooming in and out on data in the Data View. The Preview Graph appears in the Data View by default when you view logged data. When viewing live or non-logged data, right-click a display in the Data View and select **Visible Items»Preview** to display the Preview Graph. If the Data View contains multiple displays, you can display a separate Preview Graph with each display.

The blue area in the Preview Graph, which appears between two yellow cursors, indicates the section of the signal currently visible in the display. By default, the Preview Graph displays all available data. You can drag the cursors to display a subset of the data, or use the **Zoom In** and **Zoom Out** buttons to change the section of data currently displayed. You also can drag the cursors or click to the left or right of the cursors to move the cursors. If you click to the left or right of the cursor, the cursor moves to the spot you clicked. Use the scrollbar beneath the Preview Graph to scroll through the data.

Event Log

The **Event Log** tab records events such as errors, warnings, and alarms that occur in LabVIEW SignalExpress. For each event, the **Event Log** tab displays the severity of the event and the time the event occurred, as well as the source and title of the event, if known.

Double-click an event or right-click the event and select **Properties** from the shortcut menu to display an **Event Properties** window that contains more information about the event. For events caused by steps, such as errors and warnings, you can right-click the event and select **View Event Source** from the shortcut menu to display the step that caused the event. To clear the **Event Log** tab, copy events, or view only specific types of events, right-click anywhere on the tab and select the appropriate option from the shortcut menu.

If the **Event Log** tab is not visible, select **View»Event Log** to display the **Event Log** tab.

Step Setup

The **Step Setup** tab displays configuration options for steps. The **Step Setup** tab can display configuration options for only one step at a time, and the contents of the tab are specific to the step you select. Use the **Step Setup** tab to select the input signals a step processes and to set the step parameters. In some cases, the tab also includes graphs that plot the signals the step receives as inputs and returns as outputs. Click the **Preview** button that appears on the **Step Setup** tab for these steps to show or hide the graphs.

When you select a step in the Project View or add a new step to a project, LabVIEW SignalExpress automatically updates the **Step Setup** tab to display configuration options for the new step. If you want the **Step Setup** tab to always show configuration options for the currently selected step, click the **Lock to Step** button on the tab. If you lock the **Step Setup** tab to a step, LabVIEW SignalExpress opens a new **Step Setup** tab when you add or select a new step.

Project Documentation

Use the **Project Documentation** tab to document projects. For example, you can create project descriptions, display acquired data, insert images, and document measurement results. You can drag signals from the Project View to the **Project Documentation** tab to display a graph that is similar to a graph on the Data View tab. You also can print or export documentation that you create for a project. Select **View»Project Documentation** to display the **Project Documentation** tab.

Use the toolbar buttons that appear on this tab and the options in the Documentation menu to format text and objects on the tab. You can undo or redo operations on the tab using the Edit menu or keyboard shortcuts.

Recording Options

Use the **Recording Options** tab to configure data logging operations in LabVIEW SignalExpress. If a project includes steps with valid output signals, you can begin logging immediately by clicking the **Record** button and selecting a signal from the **Logging Signals Selection** dialog box. However, the **Recording Options** tab enables you to configure more advanced logging options, such as start and stop conditions, alarms, and events.

The **Recording Options** tab includes the following pages:

- Signal Selection—Specifies the signal(s) to log.
- Log Summary—Specifies the name, author, and description of a log and the folder in which to save the log file.
- Start Conditions—Specifies conditions that must occur for LabVIEW SignalExpress to start logging the signal(s).
- Stop Conditions— Specifies conditions that must occur for LabVIEW SignalExpress to stop logging the signal(s).
- Alarms—Specifies conditions under which LabVIEW SignalExpress records an alarm in the log.
- Events—Specifies occurrences that LabVIEW SignalExpress records as events in the log.

The **Recording Options** tab also includes a **Recording status** field that displays the following information about logs you are currently recording and available disk space:

- **Recording**—Indicates whether LabVIEW SignalExpress is currently logging a signal.
- **Disk Information**—Indicates how much disk space is available on the machine.
- **Current estimated log size**—Displays the estimated size of the current log in megabytes.
- **Recording time left**—Indicates how long you can continue recording the current log before you run out of memory on disk.
- **Current log started on**—Indicates the start date and time of the current log.

Signal Selection Page

Use this page of the Recording Options tab to specify which signals to log when you perform a data logging operation.

This page includes the following components:

- **Signal selection**—Displays signals you can select for logging. The project must contain at least one step with a valid output signal for a signal to appear in this list. Contains the following components:
  - **Channel Name**—Displays the names of the signals that you can record.

    Note LabVIEW SignalExpress does not support logging of frequency domain signals or timed digital signals.
  - **Record**—Specifies to record the signal.

Note You cannot use this page to select frequency-domain signals. To log a frequency-domain signal, right-click the output signal in the Project View and select **Record last value** from the shortcut menu.

Log Summary Page

Use this page of the Recording Options tab to create a summary that describes a log you create when you perform a data logging operation.

This page includes the following components:

- **Summary**—Contains the following components you can use to identify the log:
  - **Log title**—Specifies the title of the log. Enter <DATE&TIME> to name the log automatically based on the current date and time, or enter <TIME> to automatically name the log based on the current time.
  - **Author**—Specifies the name of the author of the log.
  - **Log description**—Specifies a description of the log.
  - **Prompt for title/description when log is created**—Specifies to prompt the user for a title and description of the log at the beginning of a data logging operation.
- **Log folder**—Contains the following component:
  - **Log destination folder**—Specifies the folder in which LabVIEW SignalExpress saves the log file. You also can use the **Default storage directory** field on the Logging page of the Options dialog box to specify a default log destination folder for all LabVIEW SignalExpress projects.

Start Conditions Page

Use this page of the Recording Options tab to specify which conditions must be met for data logging to start.

This page includes the following components:

- **Start condition list**—Lists the source, type, and status of each start condition you configure. Use the up and down arrows to change the order of the start conditions. Contains the following components:
  - **Source**—Indicates the source of the start condition. This field displays the **Condition type** you specify or, if you set **Condition type** to **Signal trigger**, the name of the **Signal** you specify.
  - **Conditions**—Indicates the behavior of the **Source** that triggers a start condition. This column appears empty for **Date/Time** start conditions.
  - **Met?**—Indicates whether the start condition is met.
- **Add**—Adds a start condition to the **Start condition list**.
- **Remove**—Removes the selected start condition from the **Start condition list**.
- **Condition logic**—Specifies the logic LabVIEW SignalExpress uses to process the start conditions. You must add more than one start condition to the **Start condition list** to display this pull-down menu. You can select from the following options:
  - **AND**—Specifies to start logging only when all start conditions are met.
  - **OR**—Specifies to start logging when any start condition is met.
  - **Sequence**—Specifies to start logging when all start conditions are met in the order in which they appear in the **Start condition list**. Except for the first start condition, LabVIEW SignalExpress ignores start conditions until the preceding start condition(s) is met. Use the up and down arrows next to the **Start condition list** to change the order of start conditions.
- **Condition type**—Specifies the type of the selected start

condition. You can select from the following condition types:

  - **Signal trigger**—(Default) Specifies to start logging when a trigger occurs on the **Signal** you specify. Use the **Trigger type** field to configure a trigger.
  - **Software trigger**—Specifies to start logging when a user presses the button for the **Software trigger** you specify. Software trigger buttons appear in the LabVIEW SignalExpress toolbar when you specify software triggers.
  - **Date/Time**—Specifies to start logging at the date and time you specify.

- **Signal**—[Condition type: Signal trigger] Specifies the signal to use to trigger the start condition.
- **Trigger type**—[Condition type: Signal trigger] Specifies the behavior of the **Signal** that triggers LabVIEW SignalExpress to start logging. You can select from the following options:
  - **Rising slope**—(Default) Starts logging when the signal crosses the value you specify in the **Trigger value** field with a positive slope.
  - **Falling slope**—Starts logging when the signal crosses the value you specify in the **Trigger value** field with a negative slope.
  - **Entering window**—Starts logging when the signal enters the window between the values you specify in the **Higher value** and **Lower value** fields.
  - **Leaving window**—Starts logging when the signal leaves the window between the values you specify in the **Higher value** and **Lower value** fields.
- **Trigger value**—[Trigger type: Rising slope, Falling slope] Specifies the value the signal must cross to start logging.
- **Hysteresis**—[Trigger type: Rising slope, Falling slope] Specifies the amount above or below the **Trigger value** through which the signal must pass before LabVIEW SignalExpress starts logging. You can use **Hysteresis** to prevent noise from causing a false start condition. For a **Rising slope** condition, the signal must pass below **Trigger value** – **Hysteresis** before LabVIEW SignalExpress starts logging. For a **Falling slope** condition, the

signal must pass above **Trigger value** + **Hysteresis** before LabVIEW SignalExpress starts logging.

- **Higher value**—[Trigger type: Entering window, Leaving window] Specifies the high limit of the range in which the signal must enter or leave the window to start logging.
- **Lower value**—[Trigger type: Entering window, Leaving window] Specifies the low limit of the range in which the signal must enter or leave the window to start logging.
- **Count**—[Condition type: Signal trigger] Specifies the number of times the **Signal** must meet the start condition before LabVIEW SignalExpress begins logging.
- **Software trigger**—[Condition type: Software trigger] Specifies a software trigger to use for the start condition. You can specify up to three separate software triggers. When you select a software trigger, a toolbar button with the trigger name appears. Click the toolbar button to trigger the start condition.
- **Schedule start time**—[Condition type: Date/Time] Specifies how frequently a **Date/Time** start condition occurs. You can select from the following options:
  - **Once**—(Default) Specifies for the start condition to occur at a specific time.
  - **Hourly**—Specifies for the start condition to occur hourly or in an increment of hours, minutes, and seconds that you specify.
  - **Daily**—Specifies for the start condition to occur daily or every number of days you specify.
  - **Weekly**—Specifies for the start condition to occur weekly or every number of weeks you specify on the day or days you specify.
  - **Custom**—Specifies for the start condition to occur on a custom schedule that you specify.

  Note If you set **Schedule start time** to **Hourly**, **Daily**, or **Weekly**, you must configure stop conditions that stop logging the signal between the scheduled start times.
- **Start date**—[Schedule start time: Once] Specifies the date and time to start logging the specified signal.

- **hours**—[Schedule start time: Hourly] Specifies the number of hours between each occurrence of the start condition. If you specify **minutes** and **seconds**, LabVIEW SignalExpress waits the combined total of **hours**, **minutes**, and **seconds** between each start condition.
- **minutes**—[Schedule start time: Hourly] Specifies the number of minutes between each occurrence of the start condition. If you specify **hours** and **seconds**, LabVIEW SignalExpress waits the combined total of **hours**, **minutes**, and **seconds** between each start condition.
- **seconds**—[Schedule start time: Hourly] Specifies the number of seconds between each occurrence of the start condition. If you specify **hours** and **minutes**, LabVIEW SignalExpress waits the combined total of **hours**, **minutes**, and **seconds** between each start condition.
- **Every *x* days**—[Schedule start time: Daily] Specifies the number of days between each occurrence of the start condition.
- **Every *x* week(s)**—[Schedule start time: Weekly] Specifies the number of weeks between each occurrence of the start condition. Place a checkmark in the checkbox that corresponds to the day or days on which you want LabVIEW SignalExpress to start logging.
- **Custom Schedule**—[Schedule start time: Custom] Displays a list of dates and times at which to start logging. Click the **Add** button to add a date and time to the list.
- **Add**—[Schedule start time: Custom] Displays the Set Time and Date dialog box, which you can use to add a date and time to the **Custom Schedule**.
- **Remove**—[Schedule start time: Custom] Removes the selected date and time from the **Custom Schedule**.
- **Start logging**—[Schedule start time: Hourly, Daily, Weekly] Specifies when LabVIEW SignalExpress starts logging for the first time. You can select from the following options:
    - **Immediately**—(Default) Specifies to start logging for the first time when you run the project.
    - **On date**—Specifies to start logging for the first time at the time and date you specify.

- **Start from**—[Schedule start time: Hourly, Daily, Weekly] Specifies a time from which to start logging. Select **On date** from the **Start logging** pull-down menu to enable this control.
- **Advanced timing**—Contains the following components:
    - **Pre-start condition duration (s)**—Specifies the number of seconds of data before the start condition is met to include in the data log.
    - **Start condition holdoff (s)**—Specifies a time in seconds to wait after LabVIEW SignalExpress begins acquiring a signal or after a stop condition occurs before acknowledging start conditions.
- **Restart behavior**—Contains components for configuring whether and how LabVIEW SignalExpress restarts logging after a set of start and stop conditions occur. For these components to appear, you must configure at least one stop condition. These components do not appear if you configure a **Date/Time** start condition that only occurs once. Contains the following components:
    - **Repeat start/stop cycle**—Contains the following options:
        - ***x* times**—(Default) Specifies a number of times to restart logging.
        - **Until**—Specifies a date after which LabVIEW SignalExpress no longer restarts logging.
    - **Restart start/stop cycle in**—Specifies where to save logged data after a restart occurs. You can select from the following options:
        - **current log**—(Default) Specifies to save logged data in the same log file after a restart occurs.
        - **new log**—Specifies to save logged data in a new log file after a restart occurs.

Stop Conditions Page

Use this page of the Recording Options tab to specify which conditions must be met for data logging to stop.

This page includes the following components:

- **Stop condition list**—Lists the source, type, and status of each stop condition you configure. Use the up and down arrows to change the order of the stop conditions.
    - **Source**—Indicates the source of the stop condition. This field displays the **Condition type** you specify or, if you set **Condition type** to **Signal trigger**, the name of the **Signal** you specify.
    - **Conditions**—Indicates the behavior of the **Source** that triggers a stop condition.
    - **Met?**—Indicates whether the stop condition is met.
- **Add**—Adds a stop condition to the **Stop condition list**.
- **Remove**—Removes the selected stop condition from the **Stop condition list**.
- **Condition logic**—Specifies the logic LabVIEW SignalExpress uses to process the stop conditions. You must add more than one stop condition to the **Stop condition list** to display this pull-down menu. You can select from the following options:
    - **AND**—Specifies to stop logging only when all stop conditions are met.
    - **OR**—Specifies to stop logging when any stop condition is met.
    - **Sequence**—Specifies to stop logging when all stop conditions are met in the order in which they appear in the **Stop condition list**. Except for the first stop condition, LabVIEW SignalExpress ignores stop conditions until the preceding stop condition(s) is met. Use the up and down arrows next to the **Stop condition list** to change the order of stop conditions.
- **Condition type**—Specifies the type of the selected stop condition. You can select from the following condition types:
    - **Duration**—Specifies to stop logging after an amount of

time you specify.

  - **Date/Time**—Specifies to stop logging at the date and time you specify.
  - **Signal trigger**—Specifies to stop logging when a trigger occurs on the **Signal** you specify. Use the **Trigger type** field to configure a trigger.
  - **Software trigger**—Specifies to stop logging when the **Software trigger** you specify occurs.
- **Duration (s)**—[Condition type: Duration] Specifies the number of seconds to log data before the stop condition is met.
- **Time**—[Condition type: Date/Time] Specifies the date and time to stop logging.
- **Signal**—[Condition type: Signal trigger] Specifies the signal to use to trigger the stop condition.
- **Trigger type**—[Condition type: Signal trigger] Specifies the behavior of the **Signal** that triggers LabVIEW SignalExpress to stop logging. You can select from the following options:
  - **Rising slope**—(Default) Stops logging when the signal crosses the value you specify in the **Trigger value** field with a positive slope.
  - **Falling slope**—Stops logging when the signal crosses the value you specify in the **Trigger value** field with a negative slope.
  - **Entering window**—Stops logging when the signal enters the window between the values you specify in the **Higher value** and **Lower value** fields.
  - **Leaving window**—Stops logging when the signal leaves the window between the values you specify in the **Higher value** and **Lower value** fields.
- **Trigger value**—[Trigger type: Rising slope, Falling slope] Specifies the value the signal must cross to stop logging.
- **Hysteresis**—[Trigger type: Rising slope, Falling slope] Specifies the amount above or below the **Trigger value** through which the signal must pass before LabVIEW SignalExpress stops logging. You can use **Hysteresis** to prevent noise from causing a false stop condition. For a **Rising slope** condition, the signal must

pass below **Trigger value Hysteresis** before LabVIEW SignalExpress stops logging. For a **Falling slope** condition, the signal must pass above **Trigger value** + **Hysteresis** before LabVIEW SignalExpress stops logging.

- **Higher value**—[Trigger type: Entering window, Leaving window] Specifies the high limit of the range in which the signal must enter or leave the window to stop logging.
- **Lower value**—[Trigger type: Entering window, Leaving window] Specifies the low limit of the range in which the signal must enter or leave the window to stop logging.
- **Count**—[Condition type: Signal trigger] Specifies the number of times the **Signal** must meet the stop condition before LabVIEW SignalExpress stops logging.
- **Software trigger**—[Condition type: Software trigger] Specifies a software trigger to use for the stop condition. You can specify up to three separate software triggers. When you select a software trigger, a toolbar button with the trigger name appears. Click the toolbar button to trigger the stop condition.
- **Advanced timing**—Contains the following components:
    - **Post-stop condition duration (s)**—Specifies the number of seconds of data after the stop condition is met to include in the data log.
    - **Stop condition holdoff (s)**—Specifies a time in seconds to wait after a start condition occurs before acknowledging stop conditions.

Alarms Page

Use this page of the Recording Options tab to specify alarm conditions for a data logging operation. If you specify multiple alarm conditions, an alarm activates each time the signal meets any of the alarm conditions.

This page includes the following components:

- **Alarm list**—Lists the source, type, and status of each alarm condition you configure.
- **Add**—Adds an alarm condition to the **Alarm list**.
- **Remove**—Removes the selected alarm condition from the **Alarm list**.
- **Signal**—Specifies the signal to monitor for alarm conditions.
- **Condition**—Specifies the alarm condition type that **Signal** must meet for an alarm to occur.
  - **Above**—Specifies that an alarm occurs if **Signal** is above the **Value** you specify.
  - **Below**—Specifies that an alarm occurs if **Signal** is below the **Value** you specify.
  - **Inside Range**—Specifies that an alarm occurs if **Signal** is between the **Higher value** and **Lower value** you specify.
  - **Outside Range**—Specifies that an alarm occurs if **Signal** is above the **Higher value** or below the **Lower value** you specify.
- **Value**—[Condition: Above, Below] Specifies the value **Signal** must meet for an alarm to occur.
- **Higher value**—[Condition: Inside Range, Outside Range] Specifies the high limit of the range of values **Signal** must meet for an alarm to occur.
- **Lower value**—[Condition: Inside Range, Outside Range] Specifies the lower limit of the range of values **Signal** must meet for an alarm to occur.
- **Deadband**—Specifies the deadband value for the alarm condition. The **Deadband** behavior depends on the **Condition** you select.

Condition	Deadband behavior
Above	The alarm stays active until **Signal** falls below **Value – Deadband**.
Below	The alarm stays active until **Signal** rises above **Value** + **Deadband**.
Inside Range	The alarm stays active until **Signal** falls below **Higher value** – **Deadband** or rises above **Lower value** + **Deadband**.
Outside Range	The alarm stays active until **Signal** rises above **Higher Value** + **Deadband** or falls below **Lower value** – **Deadband**.

- **Rearm time**—Specifies an amount of time that must pass between subsequent alarms.
- **Action**—Specifies the action to take when an alarm occurs.
  - **None**—Specifies to take no additional action when an alarm occurs.
  - **Beep**—Specifies to emit a system beep based on the alarm state.
  - **Display Message**—Specifies to display a user-defined message based on the alarm state.
  - **Set Digital Line**—Specifies to set the state of a digital line on a DAQmx device based on the alarm state.
  - **Set Analog Level**—Specifies to change the level of a DAQmx device analog output channel based on the alarm state.
  - **Generate Software Trigger**—Specifies to generate a software trigger based on the alarm state. You can use the software trigger to trigger a device to generate a signal.
  - **Create Snapshot**—Specifies to create a snapshot based on the alarm state.
  - **Run Program**—Specifies to execute commands based on the alarm state. For example, you can use this field to open a version of LabVIEW and run a VI.
- **Beep when alarm turns ON**—[Action: Beep] Specifies to emit a

system beep when the alarm turns on.

- **Beep when alarm turns OFF**—[Action: Beep] Specifies to emit a system beep when the alarm turns off.
- **Beep type**—[Action: Beep] Specifies the type of beep to emit. The operating system defines the sound associated with each **Beep type**. You can select from the following options:
  - **Default Beep**—Emits the OS default beep.
  - **Asterisk Beep—**Emits the OS asterisk beep.
  - **Exclamation Beep**—Emits the OS exclamation beep.
  - **Hand Beep**—Emits the OS hand beep.
  - **Question Beep**—Emits the OS question beep.
  - **OK Beep**—Emits the OS OK beep.
- **Message - Alarm ON**—[Action: Display Message] Specifies a message to display when the alarm turns on.
- **Message - Alarm OFF**—[Action: Display Message] Specifies a message to display when the alarm turns off.
- **DAQmx digital line**—[Action: Set Digital Line] Specifies the DAQmx line to toggle.
- **Action**—[Action: Set Digital Line] Specifies how to set the DAQmx line in response to the alarm. You can select from the following options:
  - **Toggle line—**Sets the DAQmx line to the same state as the alarm while the alarm is active. For example, an inactive line becomes active when the alarm asserts and returns to the inactive state when the alarm deasserts.
  - **Toggle line (inverse)**—Sets the DAQmx line to the opposite state of the alarm while the alarm is active. For example, an active line becomes inactive when the alarm asserts and returns to the active state when the alarm deasserts.
  - **HIGH when Alarm turns ON**—Sets the DAQmx line to HIGH logic when the alarm asserts. For example, an inactive line becomes active when the alarm asserts and remains in that state when the alarm deasserts.
  - **LOW when Alarm turns ON**—Sets the DAQmx line to LOW logic when the alarm asserts. For example, an

active line becomes inactive when the alarm asserts and remains in that state when the alarm deasserts.

- **Alarm/Line preview**—Displays a preview of how the **Action** you specify effects the DAQmx digital line.
- **DAQmx analog output channel**—[Action: Set Analog Level] Specifies the analog output channel of which to change the level based on the alarm state.
- **Initial level**—[Action: Set Analog Level] Specifies the initial analog channel level. LabVIEW SignalExpress sets the initial channel level when the project begins running.
- **Alarm ON - level**—[Action: Set Analog Level] Specifies the analog level when the alarm is on.
- **Alarm OFF - level**—[Action: Set Analog Level] Specifies the analog level when the alarm is off.
- **Generate trigger when alarm turns ON**—[Action: Generate Software Trigger] Specifies to generate a software trigger when the alarm turns on.
- **Trigger (ON)**—[Action: Generate Software Trigger] Specifies the software trigger to generate when the alarm turns on.
- **Generate trigger when alarm turns OFF**—[Action: Generate Software Trigger] Specifies to generate a software trigger when the alarm turns off.
- **Trigger (OFF)**—[Action: Generate Software Trigger] Specifies the software trigger to generate when the alarm turns off.
- **Create snapshot when alarm turns ON**—[Action: Create Snapshot] Specifies to create a snapshot when the alarm turns on.
- **Snapshot (ON)**—[Action: Create Snapshot] Specifies the signals to include in the snapshot. You can select from the following options:
    - **Alarmed signal**—(Default) Creates a snapshot of the signal that causes the alarm.
    - **All signals in project**—Creates a snapshot of all the signals in the project.
- **Create snapshot when alarm turns OFF**—[Action: Create Snapshot] Specifies to create a snapshot when the alarm turns

off.

- **Snapshot (OFF)**—[Action: Create Snapshot] Specifies the signals to include in the snapshot. You can select from the following options:
    - **Alarmed signal**—(Default) Creates a snapshot of the signal that causes the alarm.
    - **All signals in project**—Creates a snapshot of all the signals in the project.
- **Alarm ON Command**—[Action: Run Program] Specifies a command to execute when the alarm turns on. This field is similar to the Command Prompt in Windows. You can enter system commands and paths to executable programs. **Alarm ON Command** recognizes the following tags that you can pass as arguments to an executable program:
    - <ALARM_SIGNAL>—Specifies the name of the signal that generates the alarm.
    - <ALARM_DESCRIPTION>—Specifies the description of the alarm.
    - <ALARM_TIMESTAMP>—Specifies the timestamp at which the alarm occurs.
- **Alarm OFF Command**—[Action: Run Program] Specifies a command to execute when the alarm turns off. This field is similar to the Command Prompt in Windows. You can enter system commands and paths to executable programs. **Alarm OFF Command** recognizes the following tags that you can pass as arguments to an executable program:
    - <ALARM_SIGNAL>—Specifies the name of the signal that generates the alarm.
    - <ALARM_DESCRIPTION>—Specifies the description of the alarm.
    - <ALARM_TIMESTAMP>—Specifies the timestamp at which the alarm occurs.

Events Page

Use this page of the Recording Options tab to configure actions that LabVIEW SignalExpress recognizes as events during a data logging operation. You can configure LabVIEW SignalExpress to recognize keystroke or signal-based events.

This page includes the following components:

- **Event list**—Lists the source and type of events you configure.
- **Add**—Adds a new event to the **Event list**.
- **Remove**—Removes the selected event from the **Event list**.
- **Condition type**—Specifies the type of the selected event. You must add an event to the **Events** list to display this pull-down menu. You can select from the following options:
  - **Signal trigger**—Triggers an event when a trigger occurs on the **Signal** you specify. Use the **Trigger type** field to configure a trigger.
  - **Keystroke**—Triggers an event when the user presses the key or combination of keys you specify.
- **Signal**—[Condition type: Signal trigger] Specifies the signal to use to trigger the event.
- **Trigger type**—[Condition type: Signal trigger] Specifies the behavior of the **Signal** that triggers an event. You can select from the following options:
  - **Rising slope**—Triggers an event when the signal crosses the value you specify in the **Trigger value** field with a positive slope.
  - **Falling slope**—Triggers an event when the signal crosses the value you specify in the **Trigger value** field with a negative slope.
  - **Entering window**—Triggers an event when the signal enters the window between the values you specify in the **Higher value** and **Lower value** fields.
  - **Leaving window**—Triggers an event when the signal leaves the window between the values you specify in the **Higher value** and **Lower value** fields.
- **Trigger value**—[Trigger type: Rising slope, Falling slope]

Specifies the value the signal must cross to trigger an event.

- **Hysteresis**—[Trigger type: Rising slope, Falling slope] Specifies the amount above or below the **Trigger value** through which the signal must pass before LabVIEW SignalExpress detects the event. You can use **Hysteresis** to prevent noise from causing a false event trigger. For a **Rising slope** condition, the signal must pass below **Trigger value** – **Hysteresis** before LabVIEW SignalExpress detects the event. For a **Falling slope** condition, the signal must pass above **Trigger value** + **Hysteresis** before LabVIEW SignalExpress detects the event.
- **Higher value**—[Trigger type: Entering window, Leaving window] Specifies the high limit of the range in which the signal must enter or leave the window to trigger an event.
- **Lower value**—[Trigger type: Entering window, Leaving window] Specifies the low limit of the range in which the signal must enter or leave the window to trigger an event.
- **Key**—[Condition type: Keystroke] Specifies the key the user must press to trigger the event.
- **Modifier keys**—[Condition type: Keystroke] Specifies a key or keys that the user must press in conjunction with **Key** to trigger the event. You can select any or all of the following options:
  - **Alt**—Specifies the <Alt> key as a modifier.
  - **Ctrl**—Specifies the <Ctrl> key as a modifier.
  - **Shift**—Specifies the <Shift> key as a modifier.
  - **Windows**—Specifies the <Windows> key as a modifier.
- **Annotation**—Specifies an annotation to associate with the event. The annotation appears on the graph of the signal on the Data View tab.
- **Prompt for annotation text**—[Condition type: Keystroke] Specifies to prompt the user to enter an annotation at the time an event occurs.

Playback Options

Navigate to a Playback work area and select **View»Playback Options** to display the **Playback Options** tab.

Use the **Playback Options** tab to configure playback options for logged data. After you configure playback options, drag the logged data to the Data View tab and click the **Run** button to play back the portion of logged data you specify on the **Playback Options** tab.

Note When running logged data through analysis steps, LabVIEW SignalExpress only processes the portion of the logged data you specify on the **Playback Options** tab. If you do not configure playback options, LabVIEW SignalExpress processes the entire log by default.

- **Logged signal**—Displays all the logged signals in the project. Select a signal from the **Logged signal** list to configure playback options for that signal.
- **Start/Stop playback time**—Displays a graph of the data to play back. You can move the start and stop cursors to limit the data you want to play back.
  - **Zoom**—Zooms in on a section of the graph of data to play back. Hold the mouse button down and drag the cursor across the graph to select the section of data on which to zoom in.

    Note Zooming in on a section of data does not change the start and stop time for playback. Use the **Cursor Movement Tool** or the **Start time (s)** and **Stop time (s)** fields to change the start and stop time.
  - **Cursor Movement Tool**—Enables you to grab and move the **start** and **end** cursors on the graph to change the start and end time for playing back data. You also can right-click the graph and select **Start from here** or **End here** from the shortcut menu to move the **start** and **end** cursors.
  - **Panning Tool**—Enables you to grab the plot and move it around in the graph display.

- **Autoscale X Axis**—Autoscales the graph along the x-axis. If you zoom in on a section of data or use the **Panning Tool** to move the plot in the display, you can use this button to return the graph to its original appearance. You also can right-click the graph and select **Autoscale X axis** from the shortcut menu to autoscale the graph along the x-axis.

  Note You also can right-click the graph and select **Autoscale Y axis** from the shortcut menu to autoscale the graph along the y-axis.
- **Start time (s)**—When you select **Relative to log start time** in the **Time format** pull-down menu, specifies the time in seconds after the start of the logged signal to begin playing back the logged data.

  When you select **Absolute** in the **Time format** pull-down menu, specifies an actual timestamp from the log from which to begin playing back the logged data.
- **Stop time (s)**—When you select **Relative to log start time** in the **Time format** pull-down menu, specifies the time in seconds after the start of the logged signal to stop playing back the logged data.

  When you select **Absolute** in the **Time format** pull-down menu, specifies an actual timestamp from the log at which to stop playing back the logged data.
- **Time format**—Specifies the time format to use when configuring start and stop times for the playback of the logged data. You can select from the following options:
  - **Relative to log start time**—Configures the **Start time** and **Stop time** fields to specify the start and stop time of playback in seconds relative to the start time of the logged signal.
  - **Absolute**—Configures the **Start time** and **Stop time** fields to specify actual timestamps from the log for the start and stop time of playback.

- **Playback block configuration**—Contains options to configure the blocks of data in a logged signal. A block is a section of a signal that contains a specified number of samples. **Playback block configuration** contains the following options:

- **Block size (samples)**—Specifies the size in number of samples of each block of data during the playback operation.
- **Overlap (%)**—Specifies the percentage of overlap between blocks when playing back data.
- **Ignore last iteration if partial**—Specifies to ignore the last iteration if it contains a partial block.
- **Number of iterations**—Indicates the number of iterations LabVIEW SignalExpress plays back.
- **Sample rate**—Indicates the sample rate of the logged signal.
- **Playback preview**—Displays a preview of the playback data based on the **Block size (samples)** and **Overlap (%)**.

Menus and Toolbars

Use menu items and toolbar buttons to configure and run LabVIEW SignalExpress projects, modify the environment, and access help.

Menus

The menus at the top of the LabVIEW SignalExpress application window contain items common to other applications, such as **Open Project**, **Save Project**, **Copy**, and **Paste**, and other items specific to LabVIEW SignalExpress. Some menu items also list keyboard shortcuts. Most toolbar buttons have corresponding menu items that perform the same action.

Shortcut Menus

Some LabVIEW SignalExpress objects, including steps, graphs, and logs, have associated shortcut menus. Shortcut menus contain more object-specific options than the LabVIEW SignalExpress menus. For example, you can use the shortcut menu for the Data View to change the visible items on the **Data View** tab. To access the shortcut menu for an object, right-click the object.

Toolbars

The toolbar at the top of the LabVIEW SignalExpress application window includes buttons that perform actions on the project-level, such as running the project or displaying the Add Step palette. Some tabs also have toolbars that perform actions on the tab-level, such as adding a display to the **Data View** tab and changing the font on the Project Documentation tab.

File Menu

The **File** menu contains the following items you can use for basic file operations, such as opening, closing, and saving files:

- **New Project** creates a new project.
- **Open Project** displays a standard file dialog box you can use to navigate to and open a project.
- **Close Project** closes the current project. A confirmation dialog box prompts you to save any changes you have made to the project.
- **Revert Project** reverts the current project back to its last saved state. You cannot undo a project reversion.
- **Save Project** saves the current project. If you are saving a new project for the first time, a dialog box prompts you to name the project and determine its location.
- **Save Project As** saves the current project with a different name, with a different file type, or to a different location.
- **Import** includes the following options:
  - **Snapshots From Another Project** imports snapshots saved in LabVIEW SignalExpress project (.seproj) files.
  - **Logged Signals From Another Project** imports logged signals saved in LabVIEW SignalExpress (.seproj) files.
  - **Logged Signals from SignalExpress TDMS Files** imports logged signals saved in .tdms files.
  - **Channel View from Excel** imports data to the Channel View from a Microsoft Excel spreadsheet.
- **Export** includes the following options:
  - **Export Project Settings to XML** exports the current project configuration settings to an XML file.
  - **Copy Data View Image** copies the contents of the active Data View tab to the clipboard. You then can paste the image into other applications. This option appears only when the active tab is the **Data View** tab.
  - **Channel View to Excel** exports data from the Channel View to a Microsoft Excel spreadsheet.
  - **Export Documentation to HTML** exports documentation

from the Project Documentation tab of the current project to an HTML file. This option appears only when the active tab is the **Project Documentation** tab.

- **Page Setup** displays a **Page Setup** dialog box you can use to set print options for the application.
- **Print** appears only when the active tab is the **Data View** tab or the **Project Documentation** tab. **Print** includes the following options:
  - **Print Data View** prints the content of the **Data View** tab. This option appears only when the active tab is the **Data View** tab.
  - **Print Documentation** prints the content of the **Project Documentation** tab. This option appears only when the active tab is the **Project Documentation** tab.
- **Recent Projects** displays the most recently opened projects.
- **Exit** quits the application. A dialog box prompts you to save any changes you made to any open files before the application terminates.

Edit Menu

The **Edit** menu contains the following items that you can use to modify projects and project components:

- **Cut** contains the following options:
  - **Cut Selected Step** removes the step you select and saves it to the clipboard.
  - **Cut Text** removes the text you select and saves it to the clipboard. This option appears only when you display the Project Documentation tab.
- **Copy** contains the following options:
  - **Copy Selected Step** copies the step you select and saves it to the clipboard.
  - **Copy Text** copies the text you select and saves it to the clipboard. This option appears only when you display the **Project Documentation** tab.
- **Paste** contains the following options:
  - **Paste Before Selected Step** adds the step on the clipboard to the Project View before the selected step.
  - **Paste After Selected Step** adds the step on the clipboard to the **Project View** after the selected step.
  - **Paste Text** places the text on the clipboard on the **Project Documentation** tab. This option appears only when you display the **Project Documentation** tab.
- **Delete Selected Step** deletes the selected step.
- **Undo** cancels the last action you performed.
- **Redo** cancels the last **Undo** action you performed.

View Menu

The **View** menu contains the following items you can use to configure the appearance of the LabVIEW SignalExpress environment:

- **Reset Layout** resets the LabVIEW SignalExpress environment to the default layout.
- **Work Areas** contains the following options:
  - **Monitor / Record** displays the Monitor/Record work area.
  - **Playback** displays the Playback work area.
  - **Manage Work Areas** displays the Manage Work Areas dialog box, which you can use to create new work areas or edit or delete current work areas.
- **Channel View** displays the Channel View tab.
- **Event Log** displays the Event Log tab.
- **Operator Interface** displays the Operator Interface tab.
- **Properties** displays the Properties window.
- **Toolbox** displays the Toolbox window.
- **Data View** displays the Data View tab.
- **Recording Options** displays the Recording Options tab.
- **Project Documentation** displays the Project Documentation tab.
- **Step Setup** displays the Step Setup tab, which contains the Configuration View.
- **Show Project View** displays the Project View, which appears on the left side of the application window.

Note Additional options might appear in this menu after you install certain applications, such as NI-DAQmx.

Tools Menu

The **Tools** menu contains items that you can use to configure projects or instruments. The tools that appear in this menu depend on the assistants you have installed.

- **Generate Code** converts the current project to a LabVIEW block diagram.
- **Analyze Project** runs the Project Analyzer, which returns any errors, warnings, incompatibilities, or other issues in the current project.
- **Create NI-DAQmx Global Channel** launches the DAQ Assistant to help you create an NI-DAQmx global channel.
- **Measurement & Automation Explorer** launches the Measurement & Automation Explorer. Use the Measurement & Automation Explorer to configure hardware.
- **Refresh Available NI Devices** refreshes the list of devices you can use in hardware steps. LabVIEW SignalExpress can refresh only devices that do not require you to restart the computer in order to be detected.
- **Options** displays the **Options** dialog box.

Documentation Menu

The **Documentation** menu contains items that you can use to edit text and objects on the Project Documentation tab. The **Documentation** menu appears only when the active tab is the **Project Documentation** tab.

- **Select Font** displays the **Font** dialog box, which you can use to configure the font, style, size, color, and effects for text in the **Project Documentation** tab.
- **Align Paragraph** specifies the horizontal alignment of text and objects on the **Project Documentation** tab.
  - **Left**—Aligns text and objects with the left margin of the **Project Documentation** tab.
  - **Centered**—Aligns text and objects in the center of the **Project Documentation** tab.
  - **Right**—Aligns text and objects with the right margin of the **Project Documentation** tab.
- **Clear Project Documentation** clears the **Project Documentation** tab.
- **Insert Image** adds an image file (GIF, JPG, BMP, PNG, or EMF) to the **Project Documentation** tab.
- **Set Object Size** specifies the size of the selected object.
- **View As** contains the following options:
  - **Page View** displays the content on the **Project Documentation** tab as the content appears on a printed page.
  - **Web View** displays the content on the **Project Documentation** tab as the content appears on a Web page.

Add Step Menu

The **Add Step** menu includes the steps you can use to create projects. The steps are arranged in menus, such as the **Generate Signals** and **Create Signals** menus. Select a step from the **Add Step** menu to add it to the project.

The steps that appear in the menu depend on the assistants you have installed. Refer to the step descriptions in this help file for more information about the steps that appear in this menu.

The **Add Step** menu also appears when you right-click the Project View. You can display the **Add Step** menu as a floating window by clicking the **Add Step** button.

Operate Menu

The **Operate** menu contains the following items that you can use to configure the execution of a project:

- **Run** runs the project using the current run mode configuration.
- **Stop** stops the project when the current iteration completes.
- **Configure Run** displays the Configure Run dialog box.
- **Reset All** stops and restarts the project and resets all steps in the project to their initial states.
- **Run Continuously** runs the project continuously.
- **Run Once** runs the project for one iteration.
- **Abort** stops the project immediately, without waiting for the current iteration to complete.

Note Because aborting a project does not wait for the final iteration of the project to complete, selecting the **Abort** option might cause LabVIEW SignalExpress to display incomplete data.

- **Update Signals While Running** contains the following options:
  - **Update All Views** updates all signal views, including the Data View and previews on the Step Setup tab, when you run the project.
  - **Update None** updates none of the signal views when you run the project.
  - **Do Not Update Step Setup Tabs** updates the Data View but not the preview that appears on the **Step Setup** tab when you run the project.
  - **Set Display Update Rate** displays a dialog box you can use to specify the rate in milliseconds at which views of a signal update when you run a project.
- **Create Snapshot** displays the Create Snapshot dialog box.
- **Repeat Last Snapshot** takes a snapshot using the configuration of the last snapshot you took. This option creates a new snapshot and does not overwrite an existing snapshot.
- **Operator Mode** contains the following options:
  - **Operator Mode Enabled** enables or disables operator mode.

- **Set Operator Mode Password** displays the **Set Operator Password** dialog box, which you can use to set a password that a user must enter to disable operator mode.

Data View Menu

The **Data View** menu contains items you can use to modify the appearance of the Data View tab, create snapshots of signals displayed on the **Data View** tab, or export values of signals on the **Data View** tab to Microsoft Excel.

Note The **Data View** menu appears only when the active tab is the **Data View** tab.

This menu contains the following items:

- **Add Display** adds a display to the active **Data View** tab.
- **Rename** displays a dialog box you can use to rename the active **Data View** tab.
- **New Data View** creates a new **Data View** tab.
- **Remove Data View** removes the active **Data View** tab. This option is enabled only when the project includes more than one **Data View** tab.
- **Signals** contains the following option:
  - **Create Snapshots** creates snapshots of all the signals displayed on the active **Data View** tab.
- **Export To** contains the following option:
  - **Microsoft Excel** exports the values of every data point in every signal displayed on the active **Data View** tab to a Microsoft Excel book file. If the **Data View** tab includes multiple signals, each signal appears on a separate worksheet in the book file.

Note LabVIEW SignalExpress does not save Microsoft Excel files to which you export signals. You must manually save the exported file in Microsoft Excel.

Window Menu

The **Window** menu contains the following items that you can use to arrange the project windows and to navigate among the project windows:

- **Arrange All** stacks all open project windows.
- **Next Window** brings to the front the next project window you open.
- **Previous Window** brings to the front the previous project window you opened.

The **Window** menu also displays all open projects in the application. Select a project name to bring that project to the front.

Help Menu

The **Help** menu contains items to explain and define features of LabVIEW SignalExpress and installed assistants and to access National Instruments Technical Support.

- **LabVIEW SignalExpress Help** displays the *LabVIEW SignalExpress Help*. Use this help file as a reference for information about the LabVIEW SignalExpress environment. The *LabVIEW SignalExpress Help* also includes step-by-step instructions for using LabVIEW SignalExpress features. The *LabVIEW SignalExpress Help* also includes documentation for any assistants you install.
- **Getting Started with** contains a pull-right menu that lists a portable document format (PDF) of the Getting Started with LabVIEW SignalExpress manual and getting started guides for any other LabVIEW SignalExpress supported applications you have installed.
- **Open Example** displays a file dialog box that opens to the SignalExpress/Examples directory. Use this option to navigate to LabVIEW SignalExpress example files and projects.
- **Show Welcome Dialog** displays the **Welcome** dialog box that appears when you first launch the application.
- **Patents** displays all software and hardware patents related to National Instruments.
- **About LabVIEW SignalExpress** accesses general information about the current installation of LabVIEW SignalExpress, including version number and serial number.

Toolbar Buttons

The LabVIEW SignalExpress toolbar contains the following buttons you can use to run and edit a project or modify the LabVIEW SignalExpress environment:

Add Step displays the **Add Step** menu, where you select steps to add to the project.

Run runs the project using the current run mode configuration. When you run a project, the **Run** button becomes a **Stop** button.

Stop stops execution after the current iteration. This button appears only when you run a project.

Abort immediately stops execution without finishing the current iteration.

> **Note** In some cases, you cannot stop a project, for example when a hardware step waits on a trigger that does not occur. In this case, click the **Abort** button to stop execution.

Record opens the Logging Signals Selection dialog box.

Record While Running specifies to record the step outputs you configure to record when you click the **Run** button. This button replaces the **Record** button when you select signals to record from the Signal Selection page of the **Recording Options** tab.

Reset All simultaneously resets all steps to their initial states and restarts the processes of the steps. For example, if you have a step that averages a series of numbers, clicking the **Reset All** button restarts the averaging at 0. **Reset All** does not reset parameters to their default values.

Error List displays the error list.

Probes

Use probes to see the data a step returns without having to place the data in a graph or table in the Data View. To use probes, right-click an input or output signal in the Project View and select **Probe** from the shortcut menu to display the **Data Probe** window. The **Data Probe** window displays information about the signal. The contents of the **Data Probe** window depend on the type of signal you probe, and you can probe multiple signals at once. Each probe you set displays data in a separate **Data Probe** window.

Work Areas

Use work areas to perform multiple LabVIEW SignalExpress operations from within the same project. You can acquire data, process signals, log data, and perform measurements on logged data without opening a new project. When you save a project, LabVIEW SignalExpress saves every work area within the project in the same project file.

Selecting Work Areas

LabVIEW SignalExpress displays the current work area in a pull-down menu above the Project View. You can use this pull-down menu to navigate between work areas. Click the icon next to the menu to open the Manage Work Areas dialog box.

You also can select **View»Work Areas** to select work areas and open the **Manage Work Areas** dialog box.

Analyzing Live and Logged Data in Work Areas

Each work area contains a unique Project View and Data View. Use data logging to analyze data you acquire in multiple work areas. In the **Manage Work Areas** dialog box, select the work area in which you want to perform operations on logged data, and place a checkmark in the **Enable playback of logs** checkbox. In work areas with the **Enable playback of logs** option enabled, you can interactively scroll through data in the Data View or process logged data by playing it through analysis steps.

Creating New Work Areas

Complete the following steps to add a new work area to a project.

1. Select **View»Work Areas»Manage Work Areas**. The Manage Work Areas dialog box appears.

   Two default work areas, Monitor/Record and Playback, appear in the **Work Areas** list. You can delete or change the names and properties of these work areas, or add new work areas.
2. To add a new work area, click the **New** button.
3. Edit the name of the work area in the **Name** field.
4. Edit the description of the work area in the **Description** field.
5. If you are logging data in another work area and you want to perform operations on logged data in the new work area, place a checkmark in the **Enable playback of logs** checkbox.

   Use the **Move Up** and **Move Down** buttons to change the order of your work areas. Select a work area and click the **Delete** button to delete a work area.
6. Click the **OK** button to apply the changes and create the new work area.

Error Indicator

The error indicator appears at the bottom of the configuration view if a step encounters an error or warning. The indicator displays an error or warning icon with a short description of the error or warning. Click the **Details** button next to the short description to display the full description of the error or warning.

An icon also appears in the Project View next to the step that encountered the error or warning. You can move the cursor over the icon to view the same full description that appears when you click the **Details** button.

Project Analyzer

Select **Tools»Analyze Project** to execute the Project Analyzer. The Project Analyzer is a tool that analyzes a LabVIEW SignalExpress project and returns any errors, warnings, incompatibilities, or other issues in the Error List window. The Project Analyzer determines the task you want to complete and returns potential issues with the current LabVIEW SignalExpress configuration that can prevent the project from executing properly.

By default, LabVIEW SignalExpress executes the Project Analyzer each time you make a change in LabVIEW SignalExpress. To disable the Project Analyzer from executing each time you make a change, select **Tools»Options** to display the **Options** dialog box. On the **Execution** page, select **No** from the **Analyze Project after every change** option.

Handling Errors and Warnings

If a step encounters an error, it stops executing and all steps that process output signals from that step also stop executing. If you change the configuration of the steps to correct the error or if the state of the project changes in such a way that the error corrects itself automatically, the step that returned the error starts executing again and clears its error. Subsequent steps that inherit data from the step also start executing again.

Warnings have no effect on the execution of the project. When a step returns a warning, the step continues to execute and provide signals to subsequent steps.

Keyboard Shortcuts

The following table lists keyboard shortcuts in the LabVIEW SignalExpress environment.

Keyboard Shortcut	Description
File Operations	
Ctrl-N	Opens a new, empty project.
Ctrl-O	Opens an existing project.
Ctrl-F4	Closes the current project.
Ctrl-S	Saves the current project.
Ctrl-Shift-S	Displays a file dialog box and saves the current project using the name you specify in the dialog box.
Basic Editing	
Ctrl-X	Cuts the selected object or text.
Ctrl-C	Copies the selected object or text.
Ctrl-V	Pastes the object or text. For steps, pastes the copied step after the selected step.
Ctrl-Shift-V	Pastes a step before the selected step.
Ctrl-Z	Undoes the last action.
Ctrl-Y	Redoes the last action.
Project Execution	
Ctrl-R	Runs the project using the current run mode configuration.
Ctrl-E	Runs the project continuously.
Ctrl-Shift-R	Runs the project once.
Ctrl-Delete, Ctrl-.	Stops a running project when the current iteration completes.
Ctrl-A	Aborts a running project immediately.
Ctrl-Shift-T	Creates a snapshot.
Ctrl-T	Repeats the last snapshot.

Important Information

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

Trademarks

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

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

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

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

The Bluetooth® word mark is a registered trademark owned by the Bluetooth SIG, 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 media, or ni.com/patents.

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 award-winning National Instruments Web site at ni.com for technical support and professional services:

- Support—Technical support resources at ni.com/support include the following:
  - **Self-Help Resources**—For answers and solutions, visit ni.com/support for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on. Registered users also receive access to the NI Discussion Forums at ni.com/forums. NI Applications Engineers make sure every question submitted online receives an answer.
  - **Standard Service Program Membership**—This program entitles members to direct access to NI Applications Engineers via phone and email for one-to-one technical support, as well as exclusive access to on demand training modules via the Services Resource Center. NI offers complementary membership for a full year after purchase, after which you may renew to continue your benefits.

  For information about other technical support options in your area, visit ni.com/services or contact your local office at ni.com/contact.
- Training and Certification—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- System Integration—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.
- Declaration of Conformity (DoC)—A DoC is our claim of compliance with the Council of the European Communities using the manufacturers declaration of conformity. This system affords

the user protection for electromagnetic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting ni.com/certification.

- Calibration Certificate—If your product supports calibration, you can obtain the calibration certificate for your product at ni.com/calibration.

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

JavaScript Disabled

The HTML file you are trying to access uses JavaScript.

If you are viewing the file from your computer or from a CD or DVD and you have Internet Explorer 4.0 or later installed, JavaScript is enabled by default. If you are viewing the file from a network, such as on an intranet or on the Web, or if you do not have Internet Explorer 4.0 or later installed, you must enable JavaScript to view the file.

Display Properties Dialog Box

Right-click a display on the Data View tab and select **Properties** from the shortcut menu to display this dialog box. Use the pages of this dialog box to configure the appearance and behavior of the display. The pages that appear depend on the format of the display you are configuring.

This dialog box can contain the following pages:

- Title—Sets the title of the selected display and specifies whether LabVIEW SignalExpress shows display titles.
- Format and Precision—Sets the format and precision of numeric components of a display, such as the units of a scale on a graph axis.
- Plots—Sets the appearance of plots on a graph or chart display.
- Scales—Sets the appearance of scales and grids on graph and chart displays.
- Cursors—Sets the appearance and behavior of cursors on a graph or chart display.
- Signal Order—Sets the order of signals on a graph or chart display.
- Advanced—Sets the appearance of Boolean indicators on an LED display.

Title Page (Display Properties Dialog Box)

Click the **Properties** button on the Data View tab or right-click a display and select **Properties** from the shortcut menu to display the Display Properties dialog box. Select the **Title** tab to display this page.

Use this page to set the title of the selected display and to specify whether LabVIEW SignalExpress shows display titles on the current **Data View** tab.

This page includes the following components:

- **Display Title**—Specifies the title of the selected display.
- **Data View**—Contains the following components:
  - **Data View Title**—Specifies the title of the current **Data View** tab.
  - **Show Display Titles**—Specifies whether displays on the current **Data View** tab appear with display titles. Remove the checkmark from this checkbox to hide display titles.

Format and Precision Page (Display Properties Dialog Box)

Click the **Properties** button on the Data View tab or right-click a display and select **Properties** from the shortcut menu to display the Display Properties dialog box. Select the **Format and Precision** tab to display this page.

Use this page to configure the format and precision of a numeric display or the numeric components of other displays, such as the units of a scale on a graph axis.

This page includes the following components:

- **Component**—Specifies the numeric component, such as the axis of a graph, for which you are configuring the format and precision. This component appears only for displays that have multiple numeric components.
- **Format type**—Type of the numeric component. You can select from the following options:
  - **Floating point**—Displays the numeric component in floating-point notation.
  - **Scientific**—Displays the numeric component in scientific notation. For example, 60 in floating-point notation equals 6E+1 in scientific, where E represents the power of 10 exponent.
  - **Automatic formatting**—Displays the numeric component in the format that LabVIEW SignalExpress determines is appropriate for the data. LabVIEW SignalExpress chooses either scientific notation or floating-point notation based on the number to format.
  - **SI notation**—Displays the numeric component in System International (SI) notation, in which the unit of measurement appears after the value. For example, 6000 in floating-point notation equals 6k in SI notation.
  - **Decimal**—Displays the numeric component in base-10 decimal format.
  - **Hexadecimal**—Displays the numeric component in base-16 format. Valid digits are 0 to F. For example, 60 in

floating-point notation equals 3c in hexadecimal.

- **Octal**—Displays the numeric component in base-8 format. Valid digits are 0 to 7. For example, 60 in floating-point notation equals 74 in octal.
- **Binary**—Displays the numeric component in base-2 format. Valid digits are 0 and 1. For example, 60 in floating-point notation equals 111100 in binary.
- **Absolute time**—Displays the numeric component in terms of time elapsed since 12:00 a.m., January 1, 1904, Universal Time.
- **Relative time**—Displays the numeric component in terms of hours, minutes, and seconds starting from zero. For example, 100 in floating-point notation equals 1:40 in relative time.

- **Digits**—[Format type: Floating point, Scientific, Automatic formatting, SI notation] If **Precision Type** is **Digits of precision**, this field contains the number of digits to display after the decimal point. If **Precision Type** is **Significant digits**, this field contains the number of significant digits to display.
- **Precision Type**—[Format type: Floating point, Scientific, Automatic formatting, SI notation] Specifies whether to display digits of precision or significant digits. Select **Digits of precision** if you want the **Digits** field to indicate the number of digits to display after the decimal point. Select **Significant digits** if you want the **Digits** field to indicate the number of significant digits to display.
- **Hide trailing zeros**—[Format type: Floating point, Scientific, Automatic formatting, SI notation] Removes zeros at the end of the number. If the number has no fractional part, this option also removes the description part.
- **Exponent in multiples of 3**—[Format type: Scientific, Automatic formatting] Formats the number in engineering notation, where the exponent is always a multiple of three.
- **Use minimum field width**—[Format type: Floating point, Scientific, Automatic formatting, SI notation, Hexadecimal, Octal, Binary] Pads any excess space to the left or right of the number with zeros or spaces to reach the minimum width you enter in

Minimum field width. Place a checkmark in this checkbox to set the **Minimum field width** and **Padding**.

  - **Minimum field width**—Width to which you want to pad the number.
  - **Padding**—Sets whether to pad with spaces on the left or right or to pad with zeros on the left.

- **Time Type**—[Format type: Absolute time, Relative time] Sets the format of the time displayed in the numeric component. **Custom time format** uses the format you configure in this dialog box using the following components. **System time format** uses the format of the operating system. Select **Time unused** if you do not want to display the time in the numeric component.
  - **AM/PM type**—Specifies the type of clock to use to format the time. You can select one of the following options:
    - **AM/PM**—Specifies to use a 12-hour clock.
    - **24-hour**—Specifies to use a 24-hour clock.
  - **HMS type**—Sets whether to display hours and minutes or hours, minutes, and seconds. You can select one of the following options:
    - **HH:MM**—Specifies to display the time in hours and minutes.
    - **HH:MM:SS**—Specifies to display the time in hours, minutes, and seconds.
      - **Digits**—Specifies a number of digits to use to display fractions of seconds.
- **Date Type**—[Format type: Absolute time, Relative time] Sets the format of the date displayed in the numeric component. **Custom date format** uses the format you configure in this dialog box using the following components. **System date format** uses the format of the operating system. Select **Date unused** if you do not want to display the date in the numeric component.
  - **MDY type**—Specifies the order in which to display the month, day, and year. You can select one of the following options:
    - **M/D/Y**—Specifies to display the date in the following order: month/day/year.

- **D/M/Y**—Specifies to display the date in the following order: day/month/year.
- **Y/M/D**—Specifies to display the date in the following order: year/month/day.

– **Year type**—Sets whether to display the year and whether to display it as 2 or 4 digits. You can select one of the following options:

  - **Do not show year**—Specifies not to display the year.
  - **2-digit year**—Specifies to display the year as 2 digits.
  - **4-digit year**—Specifies to display the year as 4 digits.

Plots Page (Display Properties Dialog Box)

Click the **Properties** button on the Data View tab or right-click a display and select **Properties** from the shortcut menu to display the Display Properties dialog box. Select the **Plots** tab to display this page.

Use this page to configure the appearance of plots on a graph or chart display.

This page includes the following components:

- **Plot**—Specifies the plot you want to configure.
- **Line Style**—Specifies the line style of the plot.
- **Line Width**—Specifies the line width of the plot.
- **Point Style**—Specifies the point style of the plot.
- **Plot Interpolation**—Specifies the interpolation of the plot.
- **Colors**—Specifies the color and style of the plot line. Contains the following components:
    - **Line**—Specifies the color of the plot line.
    - **Point/fill**—Specifies the color of the point and fills.
    - **Fill to**—Specifies the baseline of the fill.
- **Show plot**—Specifies whether to display the plot on the graph. Remove the checkmark from this checkbox to hide the plot. This option is similar to the checkbox that appears next to a signal name on the graph legend.
- **Y-scale**—Specifies the y-scale to associate with the plot.

Scales Page (Display Properties Dialog Box)

Click the **Properties** button on the Data View tab or right-click a display and select **Properties** from the shortcut menu to display the Display Properties dialog box. Select the **Scales** tab to display this page.

Use this page to format scales and grids on graph and chart displays.

This page includes the following components:

- **Scale**—Specifies the scale you want to configure.
- **Name**—Specifies the name of the scale.
- **Show scale label**—Specifies whether to display the name of the scale, **Name**, on the graph or chart display.
- **Show scale**—Specifies whether to display the scale on the graph or chart display.
- **Log**—Specifies whether to map the scale logarithmically. Remove the checkmark from this checkbox to map the scale linearly.
- **Timestamp Type**—Specifies the type of timestamp to associate with the plot. You can select from the following options:
    - **Ignore**—Specifies to not associate any timestamp with the plot. LabVIEW SignalExpress displays all signals starting at time 0, regardless of when you begin acquiring the signal.
    - **Absolute**—Specifies to associate the absolute time with the plot. LabVIEW SignalExpress displays all signals starting at the absolute time at which you begin acquiring the signal.
    - **Relative To: *signal***—Specifies to plot ***signal*** starting at time 0 and to plot the other signal(s) relative to that signal. For example, if you begin acquiring the other signal(s) 10 seconds after you begin acquiring ***signal***, LabVIEW SignalExpress displays ***signal*** starting at time 0 and the other signal(s) starting at time 10.
- **X-Axis Signal**—Specifes the signal to use for the x-axis. This option appears only if the display you are configuring is an xy graph display.
- **Autoscale**—Specifies whether LabVIEW SignalExpress sets the

scale automatically based on the data in the graph. Remove the checkmark from this checkbox if you want to specify a range for the scale.

  - **Minimum**—Specifies the minimum value of the scale. Remove the checkmark from the **Autoscale** checkbox to enable this component.
  - **Maximum**—Specifies the maximum value of the scale. Remove the checkmark from the **Autoscale** checkbox to enable this component.

- **Scaling Factors**—Specify the value at the origin of the plot and adjust the scale of the plot. For example, to plot a time-domain waveform in milliseconds starting at a reference time, set **Offset** to the reference time and **Multiplier** to 0.001, because LabVIEW SignalExpress displays time in seconds by default.
  - **Offset**—Specifies the initial value for scaling data. If you change **Offset**, the scale no longer uses 0 as the origin of the plot.
  - **Multiplier**—Specifies the multiplier, or interval, for scaling data.
- **Scale Style and Colors**—Specifies the appearance of the scale. Click the button to display a pull-down menu of scale style options. This section also contains the following components for specifying the color of the scale:
  - **Major tick**—Specifies the color of the major tick mark.
  - **Minor tick**—Specifies the color of the minor tick mark.
  - **Marker text**—Specifies the color of the scale marker text.
- **Grid Style and Colors**—Specifies the appearance of the grid. Click the button to display a pull-down menu of grid style options. This section also contains the following components for specifying the color of the grid:
  - **Major grid**—Specifies the color of the major grid line.
  - **Minor grid**—Specifies the color of the minor grid line.
  - **BG Color**—Specifies the background color of the graph.

Cursors Page (Display Properties Dialog Box)

Click the **Properties** button on the Data View tab or right-click a display and select **Properties** from the shortcut menu to display the Display Properties dialog box. Select the **Cursors** tab to display this page.

Use this page to configure the appearance and behavior of cursors on a graph or chart display.

This page includes the following components:

- **Show Cursors**—Specifies whether to display cursors on the selected graph display. Place a checkmark in this checkbox to enable the components on the **Cursors** page.
- **Cursor**—Specifies the cursor you want to configure.
- **Settings**—Contains the following components:
  - **Line style**—Specifies the line style of the cursor.
  - **Line width**—Specifies the line width of the cursor.
  - **Point style**—Specifies the style of the point where the crosshairs of the cursor intersect.
  - **Crosshair style**—Specifies the style of the crosshairs of the cursor.
  - **Peak Threshold**—Specifies the value a signal must cross before LabVIEW SignalExpress recognizes peak values.
  - **Peak Width**—Specifies how far above the **Peak Threshold** value a signal must be before LabVIEW SignalExpress recognizes peak values.
  - **Cursor color**—Specifies the color of the cursor.
  - **Linked to plot**—Specifies the plot to which to link the cursor.
  - **Link Cursors**—Specifies whether to link the cursors on the selected plot to each other. If you place a checkmark in this checkbox, moving **Cursor 1** moves both cursors simultaneously. Moving **Cursor 2** moves only that cursor and changes the distance between the linked cursors.
- **Measurements**—Contains the following components for configuring the measurements that appear in the cursor legend:

- **Cursor Measurements**—Displays the measurements you can display in the cursor legend, export as step inputs, and bind to step parameters. This table includes the following columns:
    - **Name**—Displays the names of the cursor measurements.
    - **Show**—Specifies whether LabVIEW SignalExpress displays the cursor measurements in the cursor legend.
    - **Export**—Specifies whether LabVIEW SignalExpress exports the cursor measurements. If you place a checkmark in the **Export** checkbox for a cursor measurement, you can use the value of the measurement as a step input.
    - **Bind**—Specifies whether to bind the value of a cursor measurement to a step parameter. Click the button in the **Bind** column for a cursor measurement to display the **Binding Selection** dialog box. Use the **Binding Selection** dialog box to select the parameter to which to bind the cursor measurement.

  Note When you bind a cursor measurement to a step parameter, LabVIEW SignalExpress exports the cursor measurement automatically.

- **Track Minimum**—Specifies whether the cursor tracks the current minimum value of the signal plot to which you link the cursor. If you place a checkmark in this checkbox and run the project continuously, the cursor moves when LabVIEW SignalExpress detects a new minimum value.
- **Track Maximum**—Specifies whether the cursor tracks the current maximum value of the signal plot to which you link the cursor. If you place a checkmark in this checkbox and run the project continuously, the cursor moves when LabVIEW SignalExpress detects a new maximum value.

- **Linking**—Contains the following component for configuring the linking of cursors on different displays:
  - **Linking**—Lists displays with cursors to which you can

link the cursor on the display you are configuring. Place a checkmark in the checkbox next to a display name to link the cursors on the two displays. Cursors you link across displays move simultaneously when you move one of the cursors.

Signal Order Page (Display Properties Dialog Box)

Click the **Properties** button on the Data View tab or right-click a display and select **Properties** from the shortcut menu to display the Display Properties dialog box. Select the **Signal Order** tab to display this page.

Use this page to change the order of signals on a graph or chart display. This page includes the following components:

- **Signals**—Lists the signals on the graph or chart display in order from first to last.
- **Move Forward**—Moves the selected signal forward in the order. In the **Signals** list, select a signal that is not the first signal to enable this button.
- **Move Backward**—Moves the selected signal back in the order. In the **Signals** list, select a signal that is not the last signal to enable this button.
- **Move To Front**—Makes the selected signal first in the signal order. In the **Signals** list, select a signal that is not the first signal to enable this button.
- **Move To Back**—Makes the selected signal last in the signal order. In the **Signals** list, select a signal that is not the last signal to enable this button.

Advanced Page (Display Properties Dialog Box)

Click the **Properties** button on the Data View tab or right-click a display and select **Properties** from the shortcut menu to display the Display Properties dialog box. Select the **Advanced** tab to display this page.

Use this page to configure the appearance of Boolean indicators on an LED display.

This page includes the following components:

- **True color**—Specifies the color of the Boolean indicator in the on or TRUE state.
- **False color**—Specifies the color of the Boolean indicator in the off or FALSE state.

Configure Run Dialog Box

Select **Operate»Configure Run** or click the down arrow on the **Run** button and select **Configure Run** from the pull-down menu to display this dialog box.

Use this dialog box to configure the run mode for a LabVIEW SignalExpress project. This dialog box includes the following components:

- **Run the project**—Contains the following options for running the project:
  - **For *x* Iteration(s)**—Specifies the number of iterations for which you want the project to run when you click the **Run** button.
  - **For *x* Seconds**—Specifies the time in seconds for which you want the project to run when you click the **Run** button.
  - **Continuously**—Specifies to run the project continuously when you click the **Run** button.
- **Create a snapshot of all signals when the project finishes running**—Specifies whether to create a snapshot of all the signals in the project when the project finishes running.

Logging Page (Options Dialog Box)

Select **Tools»Options** to display the Options dialog box and select **Logging** from the **Category** list to display this page.

Use this page to set options for logging data in LabVIEW SignalExpress. This page includes the following components:

- **Miscellaneous**—Contains the following miscellaneous logging options:
  - **Automatically export log to ASCII file**—Specifies whether LabVIEW SignalExpress creates an ASCII text file of logged data every time you create a log. This option is similar to right-clicking a signal in the Logged Data window and selecting **Convert to ASCII** from the shortcut menu.
  - **Activate new logs**—Specifies whether a new data log becomes the active log by default.
  - **Prepare log data for viewing**—Specifies when and if LabVIEW SignalExpress prepares logged data for viewing on the Data View tab. Because most logging operations acquire numerous points of data, LabVIEW SignalExpress must process the log so you can view it on a reasonable scale in a display on the **Data View** tab. You can select from the following options:
    - **During logging**—(Default) LabVIEW SignalExpress prepares logged data for viewing as it logs the data.
    - **After logging completes**—LabVIEW SignalExpress prepares logged data for viewing after the log is complete. This option can improve logging performance on large data logs.
    - **Never**—LabVIEW SignalExpress does not prepare logged data for viewing. This option provides maximum performance benefits if you are recording a very large log, such as if you are logging for multiple days. If you want to view the data, you can right-click the log in the **Logged Data** window and select **Make Log Viewable**

from the shortcut menu.

- **Disable UI updates while logging**—Specifies whether LabVIEW SignalExpress disables updates of a signal graph on the **Data View** tab while you log the signal. Disabling UI updates can improve performance while logging.
- **Inherit playback configuration**—Specifies whether a new log inherits the settings you specify on the Playback Options tab for the previous log. By default, LabVIEW SignalExpress resets playback options each time you create a new log.
- **Restart logging on error**—Specifies whether LabVIEW SignalExpress restarts a logging operation if an error occurs.
- **Log naming convention**—Specifies the naming convention LabVIEW SignalExpress uses for new logs. You can select from the following options:
    - **Date & Time**—(Default) LabVIEW SignalExpress names new logs using the date and time at which you record the log.
    - **Time**—LabVIEW SignalExpress names new logs using the time at which you record the log.
    - **User-defined name**—LabVIEW SignalExpress names new logs using a name that you define.
- **Save copy of project next to log files**—Specifies whether LabVIEW SignalExpress saves a copy of the project you use to record a log in the directory in which you save the log.
- **Default storage directory**—Specifies the default directory in which LabVIEW SignalExpress saves log files.

  Note If you set **Store log files next to project file** to **Yes**, LabVIEW SignalExpress saves log files in the directory that contains the project regardless of the **Default storage directory** you specify.
- **Store log files next to project file**—Specifies whether

LabVIEW SignalExpress saves log files in the directory that contains the project by default. Setting this option to **Yes** overrides the **Default storage directory** you specify.

Options Dialog Box

Use this dialog box to set LabVIEW SignalExpress options. You can customize the LabVIEW SignalExpress environment as well as data handling, execution, and logging behavior. Use the **Category** list at the left side of the dialog box to set the following options:

- General—Sets options for the Add Step palette, the LabVIEW SignalExpress environment, the Event Log, and the Project View.
- Data—Sets options for exporting data from LabVIEW SignalExpress.
- Data View—Sets display options for the Data View.
- Execution—Sets project execution options.
- Logging—Sets data logging options.

Define Signal Dialog Box

Defines or edits a limit signal for the Limit Test step. In the **Configuration** page of the Limit Test step, click the **Define Upper Limit**, **Define Lower Limit**, or **Define Single Limit** buttons to display this dialog box.

You can specify the signal using a series of data points that represent the x, time or frequency, and y coordinates for the corner points of your signal. The resulting limit signal is composed of a series of straight line segments that connect these points.

This dialog box includes the following components:

- **Data Points**—Displays the values you enter to create the signal. You can enter values directly in the cells of the table or use the options in the **Rescale limit** section to create a signal.
- **Insert**—Adds a new row to the **Data Points** table.
- **Delete**—Removes the values in the **Data Points** table.
- **Rescale limit**—Contains the following options for defining values:
  - **New min. Time**—Specifies the minimum value for the x-axis scale.
  - **New max. Time**—Specifies the maximum value for the x-axis scale.
  - **New min. Ampl.**—Specifies the minimum value for the y-axis scale.
  - **New max. Ampl.**—Specifies the maximum value for the y-axis scale.
- **Load Data**—Prompts you to select a .lvm file that includes signal data you want to use to define a signal.
- **Save Data**—Saves the data you configured in **Data Points** to a .lvm file.
- **Defined Limit**—Displays the signal you define and a reference signal if you place a checkmark in the **Show input signal** checkbox.
- **Show input signal**—Displays a reference signal in the **Defined Limit** graph.
- **Show interpolated values**—Enables linear averaging and displays the interpolated values on the **Defined Upper Limit** or

Defined Lower Limit graph.

- **Frequency axis is logarithmic**—Sets the graph frequency axis to logarithmic and, when the **Limits source** is **User Defined Signals**, computes the limit values between the definition points so the resulting segment appears as a straight line in a logarithmic frequency representation. For example, you can use this to create asymptotic limits fitting filter roll-off in decibels per decade. This option is only available if the input is a frequency-domain signal.

Note If the input is a frequency-domain signal, place a checkmark in the **Frequency axis is logarithmic** checkbox to display your signal in a logarithmic frequency scale and create a limit signal connecting the corner points so that they appear as line segments in a logarithmic representation. This is useful if you want to define limits that follow an asymptotic frequency roll-off represented in decibel/decade.

Use this dialog box to define data values for a limit test.

Edit Sweep Output Dialog Box

LabVIEW SignalExpress

Specifies the Y- and X-axis parameters to accumulate during the sweep operation.

This dialog box includes the following components:

- **Output (Y-Axis)**—Lists the possible outputs to accumulate during the sweep operation.
- **Input (X-Axis)**—Lists the possible parameters to accumulate during the sweep operation.
- **Advanced**—Lists the sweep outputs you can select to represent the x-scale of the sweep operation.

Data Page (Options Dialog Box)

Select **Tools»Options** to display the Options dialog box and select **Data** from the **Category** list to display this page.

Use this page to set options for exporting data from LabVIEW SignalExpress. This page includes the following components:

- **Exporting Data**—Contains the following option for exporting data:
  - **Maximum Clipboard Data Export Size**—Specifies the maximum number of samples you can place on the clipboard to export. Increase this value if you want to export a logged signal to Microsoft Excel.

Bound Parameters List Dialog Box

Use this dialog box to bind operator interface controls to step parameters. Select a control on the Operator Interface tab, click the small arrow icon that appears to display the **Tasks** window, and click the **Edit Bound Parameters Link** to display this dialog box. You also can select a control on the **Operator Interface** tab, display the Properties window, select the Bound Parameters property that appears under **Parameter Binding**, and click the button that appears next to the value of Bound Parameters to display this dialog box.

This dialog box includes the following components:

- **Members**—Lists the step parameters that currently are bound to the control. Use the up and down arrows to change the order of parameters in the **Members** list.
- **Add**—Displays the Bound Parameter Editor dialog box, which you can use to add parameters to the **Members** list.
- **Remove**—Removes the selected parameter from the **Members** list.
- **Properties**—Lists properties of the currently selected parameter. You can edit the property values directly from the **Properties** list. The **Properties** list also includes options for scaling the values of properties.

Create Snapshot Dialog Box

Select **Operate»Create Snapshot** or press Ctrl-Shift-T to display this dialog box.

Use this dialog box to configure a snapshot of the current value(s) of a signal or signals in a project. After you configure the snapshot, click the **OK** button to create the snapshot. This dialog box includes the following components:

- **Signals to include**—Displays all the signals in the project that you can include in the snapshot. Place a checkmark in the checkbox next to a signal to include that signal in the snapshot.
- **Select All**—Selects every signal in the **Signals to include** list.
- **Select None**—Deselects all signals in the **Signals to include** list.
- **Name**—Specifies a name for the snapshot. The **Name** you specify appears in the Logged Data window when you create the snapshot.
- **Description**—Specifies a description of the snapshot. The description appears in a tip strip when you move the mouse over the name of the snapshot in the **Logged Data** window.

Manage Work Areas Dialog Box

Select **Edit»Work Areas»Manage Work Areas** to display this dialog box.

Use this dialog box to create new work areas or edit or delete current work areas.

- **Work Areas**—Lists the current work areas. **Monitor/Record** and **Playback** are the two default work areas.
- **Move Up**—Moves the selected item up in the list.
- **Move Down**—Moves the selected item down in the list.
- **Work Area Info**—Provides information for the work area selected in the **Work Areas** list.
  - **Name**—Specifies the name for the selected work area.
  - **Description**—Specifies a description of the work area.
  - **Enable playback of logs**—Specifies that you want to use the work area to analyze logged data and process logged data using analysis steps.
- **New**—Adds a new work area to the **Work Areas** list.
- **Delete**—Deletes the selected work area from the **Work Areas** list.

Error List Window

The **Error List** window displays any errors, warnings, or messages in a project. Click the **Error List** button to display the **Error List** window. The Project Analyzer also displays the **Error List** window if a project contains any errors, warnings, or messages. Select **Tools»Analyze Project** to run the Project Analyzer.

Note If a project does not contain any errors, warnings, or messages, the **Error List** button is disabled and you cannot display the **Error List** window.

Click an error, warning, or message in the top section of the **Error List** window to display a description and, if applicable, a suggested solution for the error, warning, or message.

The **Error List** window includes the following components:

- **Errors**—Displays errors in the **Error List** window.
- **Warnings**—Displays warnings in the **Error List** window.
- **Messages**—Displays messages in the **Error List** window.
- **Show Error**—Opens the Configuration View of the step that encountered the error.

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