

LabVIEW™

Getting Started with LabVIEW

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

Worldwide Offices

Australia 1800 300 800, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599,
Canada 800 433 3488, China 86 21 6555 7838, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00,
Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, India 91 80 51190000,
Israel 972 0 3 6393737, Italy 39 02 413091, Japan 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 7 095 783 68 51, Singapore 1800 226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197,
Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 02 2377 2222,
Thailand 662 278 6777, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on National Instruments documentation, refer to the National Instruments Web site at ni.com/info and enter the info code `feedback`.

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 THEREFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

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

In regards to components used in USI (Xerxes C++, ICU, and HDF5), the following copyrights apply. For a listing of the conditions and disclaimers, refer to the [USICopyrights.chm](#).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright © 1999 The Apache Software Foundation. All rights reserved.

Copyright © 1995–2003 International Business Machines Corporation and others. All rights reserved.

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities

Copyright © 1998, 1999, 2000, 2001, 2003 by the Board of Trustees of the University of Illinois. All rights reserved.

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 trademark of Apple Computer, Inc., registered in the U.S. and other countries. Other product and company names mentioned herein are trademarks or trade names of their respective companies.

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

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or 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.

Contents

About This Manual

Conventions	ix
Related Documentation.....	x

Chapter 1

Getting Started with LabVIEW Virtual Instruments

Building a Virtual Instrument	1-1
Launching LabVIEW	1-2
Opening a New VI from a Template	1-3
Adding a Control to the Front Panel.....	1-5
Changing a Signal Type	1-6
Wiring Objects on the Block Diagram	1-8
Running a VI	1-9
Modifying a Signal	1-10
Displaying Two Signals on a Graph.....	1-12
Customizing a Knob Control.....	1-13
Customizing a Waveform Graph.....	1-15
Summary	1-16
New Dialog Box and Template VIs	1-16
Front Panel.....	1-16
Property Dialog Boxes	1-17
Block Diagram.....	1-17
Express VIs.....	1-17

Chapter 2

Analyzing and Saving a Signal

Building a VI from a Template	2-1
Opening a New VI from a Template	2-1
Modifying the Block Diagram.....	2-3
Modifying the Front Panel.....	2-4
Analyzing the Amplitude of a Signal	2-5
Adding a Warning Light.....	2-6
Setting a Warning Level Limit.....	2-7
Warning the User.....	2-8
Configuring a VI to Save Data to a File	2-9
Saving Data to a File	2-10
Adding a Button That Stores Data When Pressed.....	2-10

Saving Data When Prompted by a User.....	2-11
Viewing Saved Data.....	2-12
Summary.....	2-13
LabVIEW Documentation Resources	2-13
Controls and Indicators	2-13
Saving Data	2-13
Errors and Broken Wires.....	2-14

Chapter 3 Customizing a VI

Building a VI from a Blank VI.....	3-1
Opening a Blank VI	3-1
Adding an Express VI That Simulates a Signal.....	3-2
Searching the Help and Modifying a Signal	3-3
Customizing a User Interface from the Block Diagram	3-4
Configuring a VI to Run Continuously until the User Stops It.....	3-5
Searching for Examples	3-6
Controlling the Speed of Execution	3-7
Using a Table to Display Data	3-8
Summary.....	3-9
Using the LabVIEW Help Resources	3-9
Customizing the Block Diagram Code	3-10
Creating Controls and Indicators	3-10
Controlling When a VI Stops Running.....	3-11
Displaying Data in a Table	3-11

Chapter 4 Optional: Acquiring Data and Communicating with Instruments

Acquiring a Signal.....	4-1
Creating an NI-DAQmx Task	4-2
Testing the Task	4-4
Graphing Data from a DAQ Device	4-4
Editing an NI-DAQmx Task	4-5
Visually Comparing Two Voltage Readings	4-6
Communicating with an Instrument	4-6
Selecting an Instrument.....	4-7
Acquiring and Parsing Information for an Instrument.....	4-7
Summary.....	4-8
DAQ Assistant Express VI	4-8
Tasks	4-9
Instrument I/O Assistant Express VI	4-9

Chapter 5

Using Other LabVIEW Features

All Controls and Indicators	5-1
All VIs and Functions	5-2
VIs	5-2
Functions	5-2
Data Types	5-3
Dynamic Data Type.....	5-3
When to Use Other LabVIEW Features	5-4

Appendix A

Technical Support and Professional Services

Glossary

Index

About This Manual

Use this manual as a tutorial to familiarize yourself with the LabVIEW graphical programming environment and the basic LabVIEW features you use to build data acquisition and instrument control applications.

This manual contains exercises that you can use to learn how to develop basic applications in LabVIEW. These exercises take a short amount of time to complete and help you get started with LabVIEW.

The end of each chapter includes a summary of the main concepts taught in that chapter. Use these summaries to review what you learned.

Conventions

The following conventions appear in this manual:

»

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.

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, controls and indicators on the front panel, window names, dialog boxes, sections of dialog boxes, menu names, and palette names.

italic

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

monospace

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

monospace bold	Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.
Platform	Text in this font denotes a specific platform and indicates that the text following it applies only to that platform.
right-click	(Mac OS) Press <Command>-click to perform the same action as a right-click.

Related Documentation

The following documents contain information that you may find helpful as you read this manual:

- *LabVIEW Release Notes*—Use these release notes to install and uninstall LabVIEW. The release notes also describe the system requirements for the LabVIEW software, including the LabVIEW Application Builder.
- *LabVIEW Help*—Use this help file to access information about LabVIEW programming concepts, step-by-step instructions for using LabVIEW, and reference information about LabVIEW VIs, functions, palettes, menus, and tools. Access the *LabVIEW Help* by selecting **Help»Search the LabVIEW Help**.
- *LabVIEW Quick Reference Card*—Use this card as a reference for information about help resources, keyboard shortcuts, data type terminals, and tools for editing, execution, and debugging.

Getting Started with LabVIEW Virtual Instruments

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. LabVIEW contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data, as well as tools to help you troubleshoot code you write.

In LabVIEW, you build a user interface, or front panel, with controls and indicators. Controls are knobs, push buttons, dials, and other input mechanisms. Indicators are graphs, LEDs, and other output displays. After you build the user interface, you add code using VIs and structures to control the front panel objects. The block diagram contains this code.

You can use LabVIEW to communicate with hardware such as data acquisition, vision, and motion control devices, as well as GPIB, PXI, VXI, RS232, and RS485 instruments.

Building a Virtual Instrument

In the following exercises, you will build a VI that generates a signal and displays that signal in a graph. When you complete the exercises, the front panel of the VI will look similar to the front panel in Figure 1-1.



You can complete the exercises in this chapter in approximately 40 minutes.

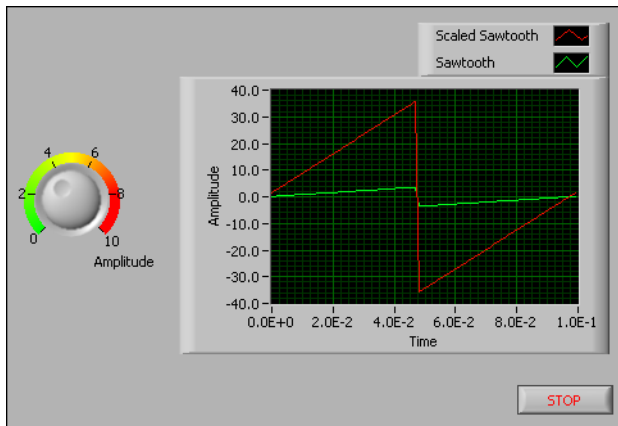


Figure 1-1. Front Panel of the Acquiring a Signal VI

Launching LabVIEW

The **Getting Started** window, shown in Figure 1-2, appears when you launch LabVIEW. Use this window to create new VIs, select among the most recently opened LabVIEW files, find examples, and launch the *LabVIEW Help*. You also can access information and resources to help you learn about LabVIEW, such as specific manuals, help topics, and resources on the National Instruments Web site, ni.com.

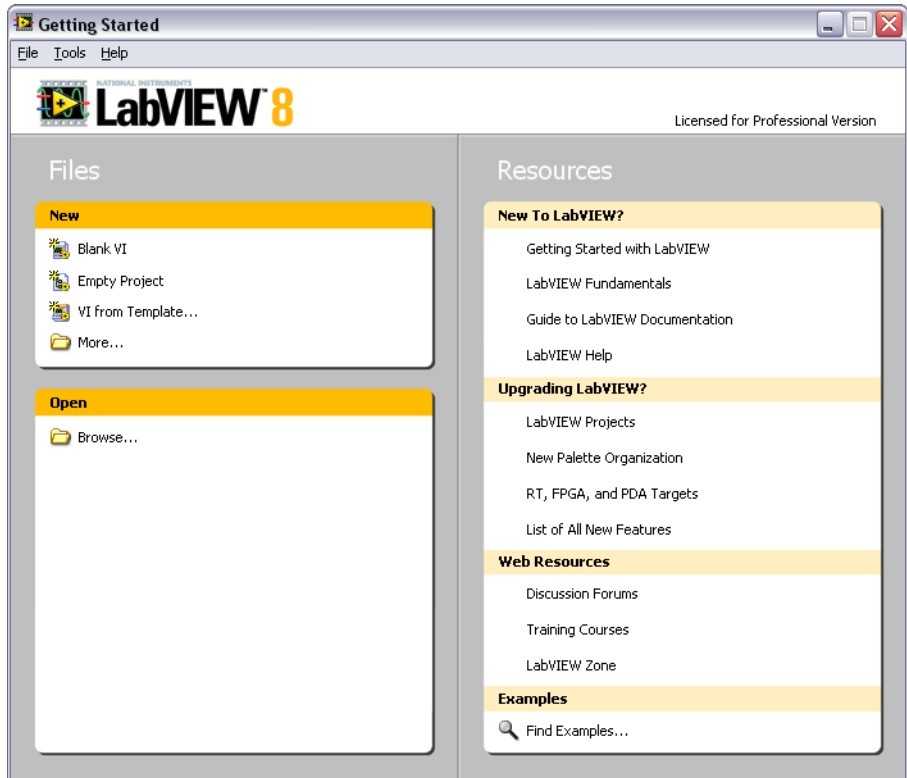


Figure 1-2. Getting Started Window

The **Getting Started** window disappears when you open an existing file or create a new file. The **Getting Started** window appears when you close all open front panels and block diagrams. You also can display the window by selecting **View»Getting Started Window**.

Opening a New VI from a Template

LabVIEW provides built-in template VIs that include the subVIs, functions, structures, and front panel objects you need to get started building common measurement applications.

Complete the following steps to create a VI that generates a signal and displays it on the front panel.

1. Launch LabVIEW.
2. In the **Getting Started** window, click the **New** or **VI from Template** link to display the **New** dialog box.

- From the **Create New** list, select **VI»From Template»Tutorial (Getting Started)»Generate and Display**. This template VI generates and displays a signal.

A preview and a brief description of the template VI appear in the **Description** section. Figure 1-3 shows the **New** dialog box and the preview of the Generate and Display template VI.

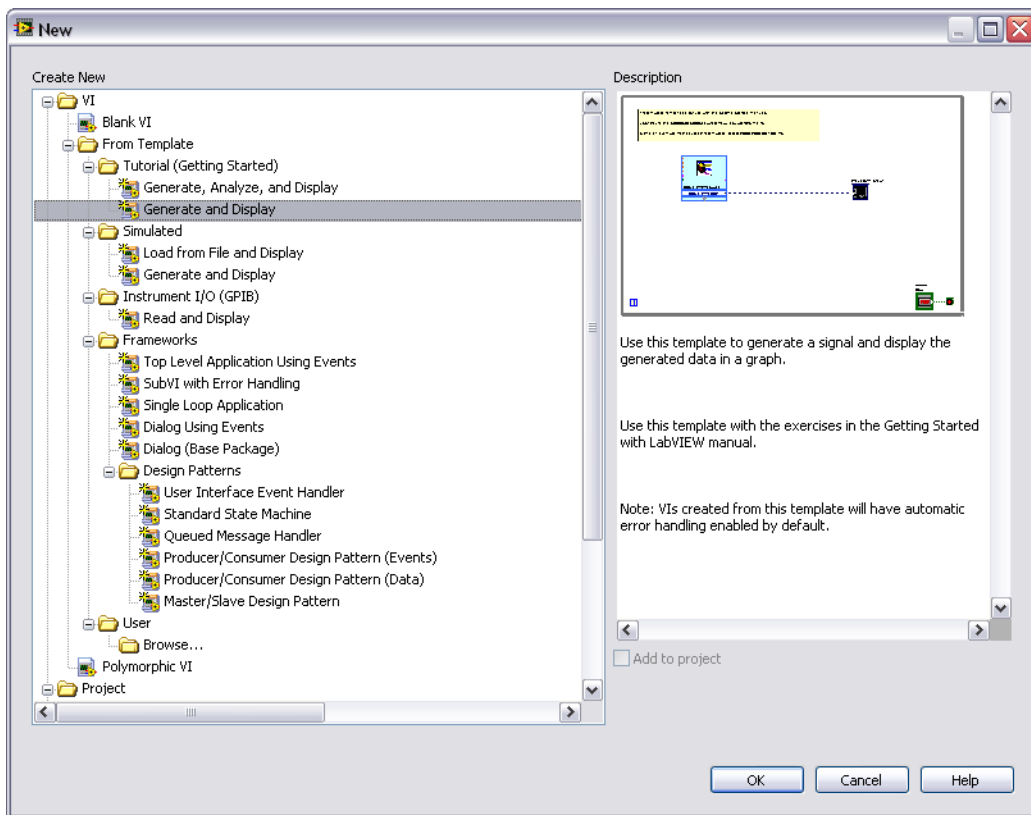


Figure 1-3. New Dialog Box

- Click the **OK** button to create a VI from the template. You also can double-click the name of the template VI in the **Create New** list to create a VI from the template.

5. Examine the front panel of the VI.

The user interface, or front panel, appears with a gray background and includes controls and indicators. The title bar of the front panel indicates that this window is the front panel for the Generate and Display VI.



Note If the front panel is not visible, you can display the front panel by selecting **Window»Show Front Panel**.

6. Select **Window»Show Block Diagram** and examine the block diagram of the VI.

The block diagram appears with a white background and includes VIs and structures that control the front panel objects. The title bar of the block diagram indicates that this window is the block diagram for the Generate and Display VI.



7. On the front panel toolbar, click the **Run** button, shown at left. You also can press the <Ctrl-R> keys to run a VI.



Note The <Ctrl> key in keyboard shortcuts corresponds to the (**Mac OS**) <Option> or <Command> key or (**Linux**) <Alt> key.

A sine wave appears on the graph on the front panel.



8. Stop the VI by clicking the **STOP** button, shown at left, on the front panel.

Adding a Control to the Front Panel

Controls on the front panel simulate the input mechanisms on a physical instrument and supply data to the block diagram of the VI. Many physical instruments have knobs you can turn to change an input value.

Complete the following steps to add a knob control to the front panel.



Tip Throughout these exercises, you can undo recent edits by selecting **Edit»Undo** or pressing the <Ctrl-Z> keys.

1. If the **Controls** palette, shown in Figure 1-4, is not visible on the front panel, select **View»Controls Palette** to display it. The **Controls** palette opens with the **Express** subpalette visible by default. If you have selected another subpalette, you can return to the **Express** subpalette by clicking **Express** on the **Controls** palette.

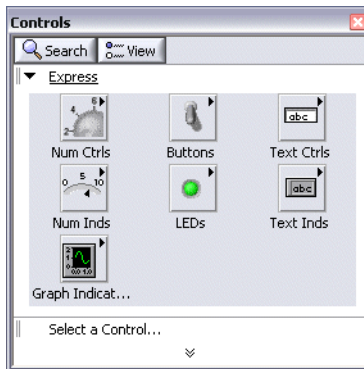


Figure 1-4. Controls Palette

2. Move the cursor over the icons on the **Controls** palette to locate the **Numeric Controls** palette.

When you move the cursor over icons on the **Controls** palette, the name of the subpalette, control, or indicator appears in a tip strip below the icon.

3. Click the **Numeric Controls** icon to display the **Numeric Controls** palette.
4. Click the knob control on the **Numeric Controls** palette to attach the control to the cursor, then place the knob on the front panel to the left of the waveform graph.

You will use this knob in a later exercise to control the amplitude of a signal.

5. Select **File»Save As** and save the VI as `Acquiring a Signal.vi` in an easily accessible location.

Changing a Signal Type

The block diagram has a blue icon labeled **Simulate Signal**. This icon represents the Simulate Signal Express VI. The Simulate Signal Express VI simulates a sine wave by default.

Complete the following steps to change this signal to a sawtooth wave.

1. Display the block diagram by pressing the `<Ctrl-E>` keys or by clicking the block diagram. Pressing the `<Ctrl-E>` keys switches from the front panel to the block diagram or from the block diagram to the front panel.



Locate the Simulate Signal Express VI, shown at left. An Express VI is a component of the block diagram that you can configure to perform common measurement tasks. The Simulate Signal Express VI simulates a signal based on the configuration that you specify.

2. Right-click the Simulate Signal Express VI and select **Properties** from the shortcut menu to display the **Configure Simulate Signal** dialog box. (**Mac OS**) Press <Control>-click to perform the same action as right-click.

You also can double-click the Express VI to display the **Configure Simulate Signal** dialog box. If you wire data to an Express VI and run it, the Express VI displays real data in the configuration dialog box. If you close and reopen the Express VI, the VI displays sample data in the configuration dialog box until you run the VI again.

3. Select **Sawtooth** from the **Signal type** pull-down menu.

The waveform on the graph in the **Result Preview** section changes to a sawtooth wave. The **Configure Simulate Signal** dialog box should appear similar to Figure 1-5.

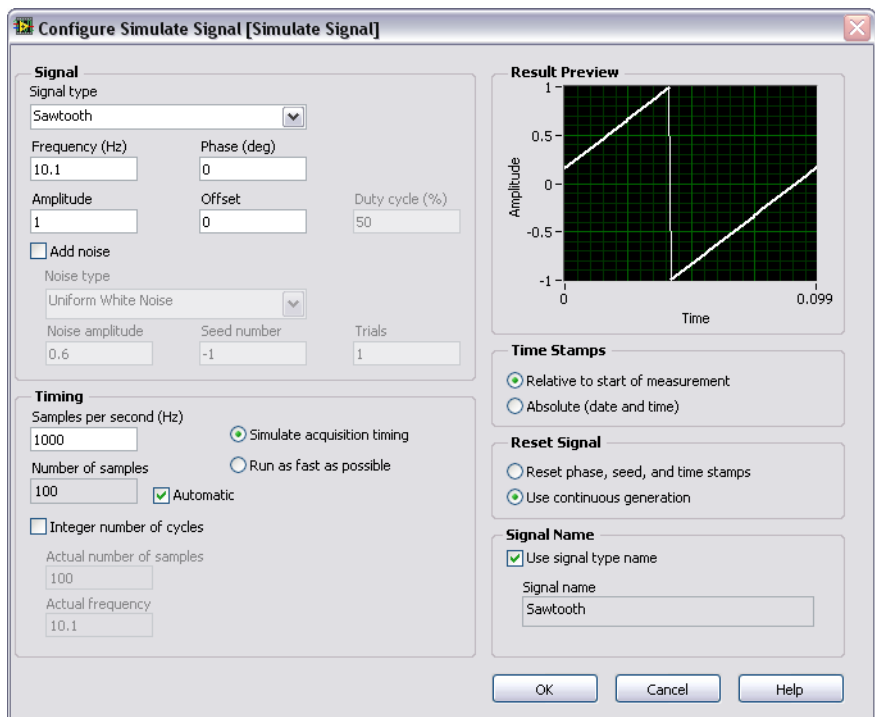


Figure 1-5. Configure Simulate Signal Dialog Box

4. Click the **OK** button to save the current configuration and close the **Configure Simulate Signal** dialog box.
5. Move the cursor over the down arrows at the bottom of the Simulate Signal Express VI. The down arrows indicate you can reveal hidden content by extending the border of the Express VI.
6. When a double-headed arrow appears, shown at left, click and drag the border of the Express VI to add two rows. When you release the border, the **Amplitude** input appears.



Because the **Amplitude** input appears on the block diagram, you can configure the amplitude of the sawtooth wave on the block diagram.

In Figure 1-5, notice that **Amplitude** is an option in the **Configure Simulate Signal** dialog box. When inputs, such as **Amplitude**, appear on the block diagram and in the configuration dialog box, you can configure the inputs in either location.

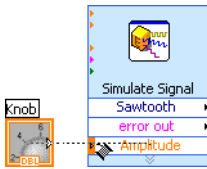
Wiring Objects on the Block Diagram

To use the knob control to change the amplitude of the signal, you must connect two objects on the block diagram.

Complete the following steps to wire the knob to the **Amplitude** input of the Simulate Signal Express VI.



1. On the block diagram, move the cursor over the **Knob** terminal, shown at left.
The cursor becomes an arrow, or the Positioning tool, shown at left. Use the Positioning tool to select, position, and resize objects.
2. Use the Positioning tool to select the **Knob** terminal and make sure it is to the left of the Simulate Signal Express VI and inside the gray structure, shown at left.
The terminals inside the loop are representations of front panel controls and indicators. Terminals are entry and exit ports that exchange information between the front panel and block diagram.
3. Deselect the **Knob** terminal by clicking a blank space on the block diagram. You must deselect an object to switch from using the Positioning tool with the object to another tool.
4. Move the cursor over the arrow on the **Knob** terminal, shown at left.
The cursor becomes a wire spool, or the Wiring tool, shown at left. Use the Wiring tool to wire objects together on the block diagram.



- When the Wiring tool appears, click the arrow on the **Knob** terminal and then click the arrow on the **Amplitude** input of the Simulate Signal Express VI, shown at left, to wire the two objects together.

A wire appears and connects the two objects. Data flows along this wire from the **Knob** terminal to the Express VI.

- Select **File»Save** to save the VI.

Running a VI

Running a VI executes the solution. Complete the following steps to run the Acquiring a Signal VI.

- Display the front panel by pressing the <Ctrl-E> keys or by clicking the front panel.
- Click the **Run** button or press the <Ctrl-R> keys to run the VI.
- Move the cursor over the knob control.



The cursor becomes a hand, or the Operating tool, shown at left. Use the Operating tool to change the value of a control.

- Using the Operating tool, turn the knob to adjust the amplitude of the sawtooth wave.

The amplitude of the sawtooth wave changes as you turn the knob. The y-axis on the graph scales automatically to account for the change in amplitude.



To indicate that the VI is running, the **Run** button changes to a darkened arrow, shown at left. You can change the value of most controls while a VI runs, but you cannot edit the VI in other ways while the VI runs.

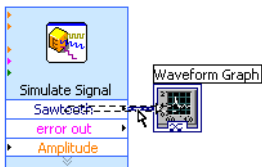
- Click the **STOP** button, shown at left, to stop the VI.



The **STOP** button stops the VI after the VI completes the current iteration. The **Abort Execution** button, shown at left, stops the VI immediately, before the VI finishes the current iteration. Aborting a VI that uses external resources, such as external hardware, might leave the resources in an unknown state by not resetting or releasing them properly. Design VIs with a stop button to avoid this problem.

Modifying a Signal

Complete the following steps to add scaling to the signal and display the results in the graph on the front panel.



1. On the block diagram, use the Positioning tool to double-click the wire that connects the Simulate Signal Express VI to the **Waveform Graph** terminal, shown at left.
2. Press the <Delete> key to delete this wire.
3. If the **Functions** palette, shown in Figure 1-6, is not visible, select **View»Functions Palette** to display it. The **Functions** palette opens with the **Express** subpalette visible by default. If you have selected another subpalette, you can return to the **Express** subpalette by clicking **Express** on the **Functions** palette.

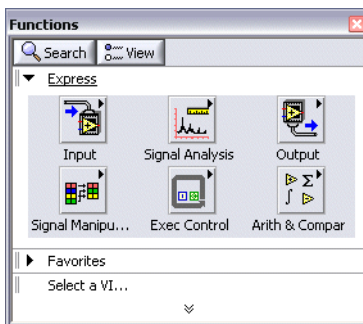


Figure 1-6. Functions Palette



4. On the **Arithmetic & Comparison** palette, select the Scaling and Mapping Express VI, shown at left, and place it on the block diagram inside the loop between the Simulate Signal Express VI and the **Waveform Graph** terminal. You can move the **Waveform Graph** terminal to the right to make more room between the Express VI and the terminal.



Note The Scaling and Mapping Express VI is available only in the LabVIEW Full and Professional Development Systems. If you are using the Base Package, skip to the next section.

The **Configure Scaling and Mapping** dialog box opens when you place the Express VI on the block diagram.

- Define the value of the scaling factor by entering 10 in the **Slope (m)** text box.

The **Configure Scaling and Mapping** dialog box should appear similar to Figure 1-7.

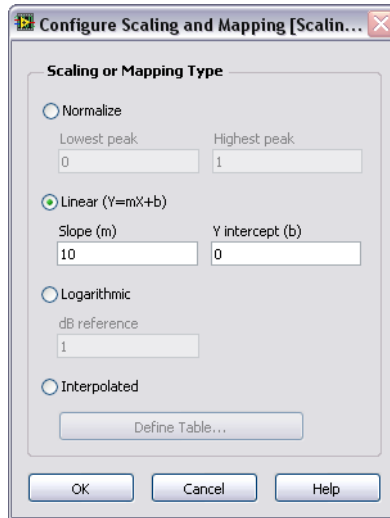
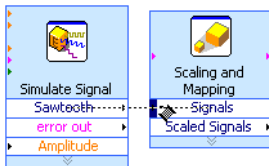


Figure 1-7. Configure Scaling and Mapping Dialog Box

- Click the **OK** button to save the current configuration and close the **Configure Scaling and Mapping** dialog box.
- Move the cursor over the arrow on the **Sawtooth** output of the Simulate Signal Express VI.
- When the Wiring tool appears, click the arrow on the **Sawtooth** output and then click the arrow on the **Signals** input of the Scaling and Mapping Express VI, shown at left, to wire the two objects together.



- Use the Wiring tool to wire the **Scaled Signals** output of the Scaling and Mapping Express VI to the **Waveform Graph** terminal.

Examine the wires connecting the Express VIs and terminals. The arrows on the Express VIs and terminals indicate the direction that the data flows along these wires. The block diagram should appear similar to Figure 1-8.

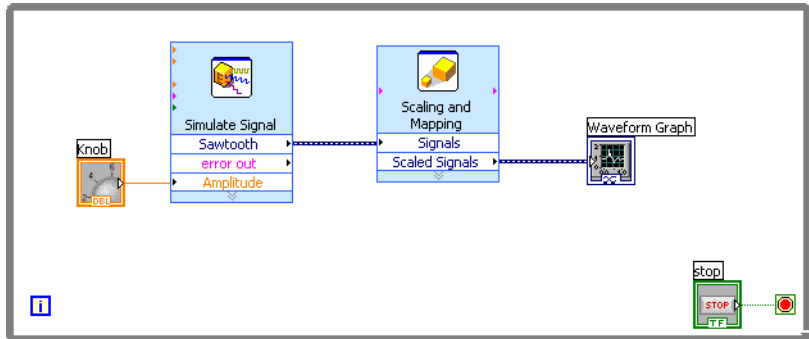


Figure 1-8. Block Diagram of the Acquiring a Signal VI

- Press the <Ctrl-S> keys or select **File»Save** to save the VI.

Displaying Two Signals on a Graph

To compare the signal generated by the Simulate Signal Express VI and the signal modified by the Scaling and Mapping Express VI on the same graph, use the Merge Signals function.

Complete the following steps to display two signals on the same graph.

- On the block diagram, move the cursor over the arrow on the **Sawtooth** output of the Simulate Signal Express VI.
- Use the Wiring tool to wire the **Sawtooth** output to the **Waveform Graph** terminal.



The Merge Signals function, shown at left, appears where the two wires connect. A function is a built-in execution element, comparable to an operator, function, or statement in a text-based programming language. The Merge Signals function takes the two separate signals and combines them so that both can display on the same graph.

The block diagram should appear similar to Figure 1-9.

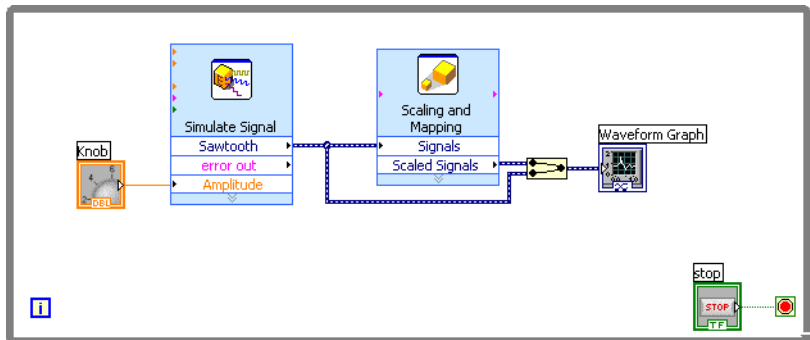


Figure 1-9. Block Diagram Showing the Merge Signals Function

3. Press the <Ctrl-S> keys or select **File»Save** to save the VI.
4. Return to the front panel, run the VI, and turn the knob control.

The graph plots the sawtooth wave and the scaled signal. The maximum value on the y-axis automatically changes to be 10 times the knob value. This scaling occurs because you set the slope to 10 in the Scaling and Mapping Express VI.

5. Click the **STOP** button to stop the VI.

Customizing a Knob Control

The knob control changes the amplitude of the sawtooth wave, so labeling it **Amplitude** accurately describes the behavior of the knob.

Complete the following steps to customize the appearance of the knob control.

1. On the front panel, right-click the knob and select **Properties** from the shortcut menu to display the **Knob Properties** dialog box.
2. In the **Label** section on the **Appearance** page, delete the label **knob**, and enter **Amplitude** in the text box.

The **Knob Properties** dialog box should appear similar to Figure 1-10.

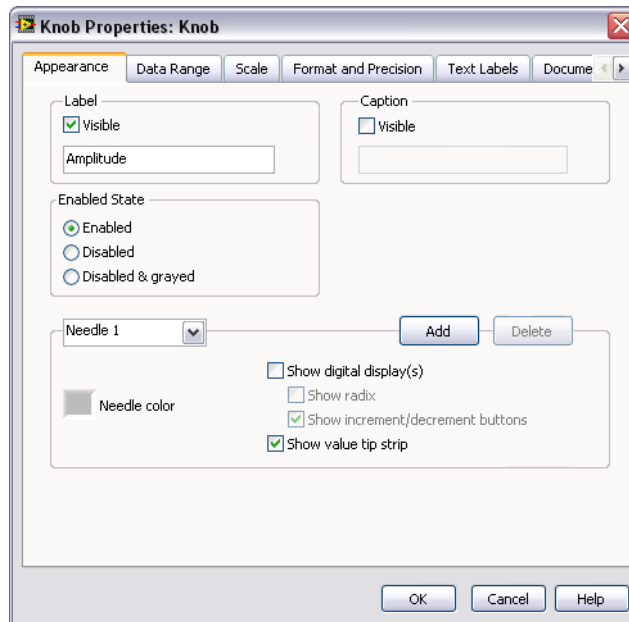


Figure 1-10. Knob Properties Dialog Box

3. Click the **Scale** tab and in the **Scale Style** section, place a checkmark in the **Show color ramp** checkbox.
The knob on the front panel updates to reflect these changes.
4. Click the **OK** button to save the current configuration and close the **Knob Properties** dialog box.
5. Save the VI.
6. Reopen the **Knob Properties** dialog box and experiment with other properties of the knob. For example, on the **Scale** page, try changing the colors for the **Marker text color** by clicking the color box.
7. Click the **Cancel** button to avoid applying any changes you made while experimenting. If you want to keep the changes you made, click the **OK** button.

Customizing a Waveform Graph

The waveform graph indicator displays the two signals. To indicate which plot is the scaled signal and which is the simulated signal, you can customize the plots.

Complete the following steps to customize the appearance of the waveform graph indicator.

1. On the front panel, move the cursor over the top of the plot legend on the waveform graph.

Though the graph has two plots, the plot legend displays only one plot.

2. When a double-headed arrow appears, shown in Figure 1-11, click and drag the border of the plot legend until the second plot name appears.

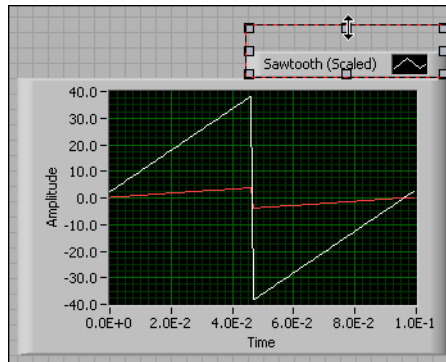


Figure 1-11. Expanding a Plot Legend

3. Right-click the waveform graph and select **Properties** from the shortcut menu to display the **Waveform Graph Properties** dialog box.
4. On the **Plots** page, select **Sawtooth** from the pull-down menu. In the **Colors** section, click the **Line** color box to display the color picker. Select a new line color.
5. Select **Sawtooth (Scaled)** from the pull-down menu.
6. Place a checkmark in the **Do not use waveform names for plot names** checkbox.
7. In the **Name** text box, delete the current label and change the name of this plot to `Scaled Sawtooth`.

8. Click the **OK** button to save the current configuration and close the **Waveform Graph Properties** dialog box.
The plot color on the front panel changes.
9. Reopen the **Waveform Graph Properties** dialog box and experiment with other properties of the graph. For example, on the **Scales** page, try disabling automatic scaling and changing the minimum and maximum value of the y-axis.
10. Click the **Cancel** button to avoid applying any changes you made while experimenting. If you want to keep the changes you made, click the **OK** button.
11. Save and close the VI.

Summary

The following topics are a summary of the main concepts you learned in this chapter.

New Dialog Box and Template VIs

The **New** dialog box contains many template VIs, including the ones used in this manual. The template VIs help you start building VIs for common measurements and other tasks. The template VIs include the Express VIs, functions, and front panel objects you need to get started building common measurement applications.

Use one of the following methods to access the **New** dialog box:

- Click the **New, VI from Template, or More** links in the **Getting Started** window after you start LabVIEW.
- Select **File»New** from the menu bar of the **Getting Started** window, the front panel, or the block diagram.

Front Panel

The front panel is the user interface of a VI. You build the front panel by using controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls and indicators are located on the **Controls** palette.

Controls are knobs, push buttons, dials, and other input mechanisms. Controls simulate instrument input mechanisms and supply data to the block diagram of the VI.

Indicators are graphs, LEDs, and other displays. Indicators simulate instrument output mechanisms and display data the block diagram acquires or generates.

Property Dialog Boxes

Use property dialog boxes or shortcut menus to configure how controls and indicators appear or behave on the front panel. Right-click a control or indicator on the front panel and select **Properties** from the shortcut menu to access the property dialog box for that object. You cannot access property dialog boxes for a control or indicator when a VI is running.

Block Diagram

The block diagram contains the graphical source code, also known as G code or block diagram code, for how the VI runs. The block diagram code uses graphical representations of functions to control the front panel objects. Front panel objects appear as icon terminals on the block diagram. Wires connect control and indicator terminals to Express VIs, VIs, and functions. Data flows through the wires from controls to VIs and functions, from VIs and functions to other VIs and functions, and from VIs and functions to indicators. The movement of data through the nodes on the block diagram determines the execution order of the VIs and functions. This movement of data is known as dataflow programming.

Express VIs

Use Express VIs located on the **Functions** palette for common measurement tasks. When you place an Express VI on the block diagram, the dialog box you use to configure that Express VI appears by default. Set the options in this configuration dialog box to specify how the Express VI behaves. You also can double-click an Express VI or right-click an Express VI and select **Properties** from the shortcut menu to display the configuration dialog box. If you wire data to an Express VI and run it, the Express VI displays real data in the configuration dialog box. If you close and reopen the Express VI, the VI displays sample data in the configuration dialog box until you run the VI again.

Express VIs appear on the block diagram as expandable nodes with icons surrounded by a blue field. You can resize an Express VI to display its inputs and outputs. The inputs and outputs you can display for the Express VI depend on how you configure the VI.

Analyzing and Saving a Signal

LabVIEW includes a set of Express VIs that help you analyze signals. This chapter teaches you how to use LabVIEW to perform a basic analysis of a signal and how to save the analyzed data to a file.

Building a VI from a Template

In the following exercises, you will build a VI that generates a signal, extracts the DC value of the signal, indicates if the signal exceeds a certain limit, and records the data. When you complete the exercises, the front panel of the VI will look similar to the front panel in Figure 2-1.



You can complete the exercises in this chapter in approximately 40 minutes.

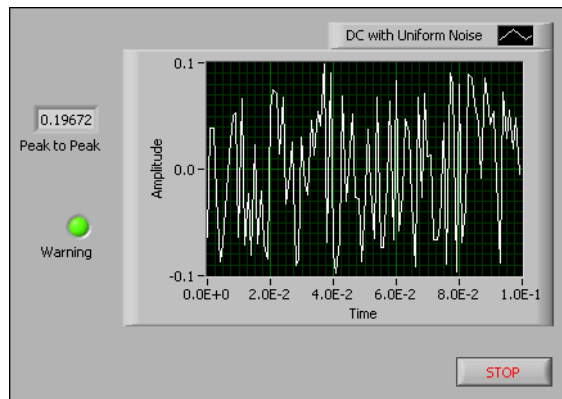


Figure 2-1. Front Panel of the Warning Light VI

Opening a New VI from a Template

To build this VI, you can start from the **New** dialog box.

Complete the following steps to select a new template VI that generates, analyzes, and displays a signal.

1. In the **Getting Started** window, click the **New** link to display the **New** dialog box.
2. From the **Create New** list, select **VI»From Template»Tutorial (Getting Started)»Generate, Analyze, and Display**. This template VI simulates a signal and analyzes it for its root mean square (RMS) value.
3. Click the **OK** button to create a VI from the template. You also can double-click the name of the template VI in the **Create New** list to create a VI from the template.
4. If the **Context Help** window, shown in Figure 2-2, is not visible, select **Help»Show Context Help** from the front panel or block diagram menu bar to display the **Context Help** window.



Tip You also can press the <Ctrl-H> keys to display the **Context Help** window.

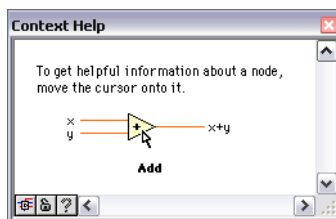
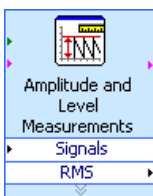


Figure 2-2. Context Help Window

5. Display the block diagram by pressing the <Ctrl-E> keys.
6. Move the cursor over the Amplitude and Level Measurements Express VI, shown at left.



When you move the cursor over the Express VI, the **Context Help** window displays information about the Express VI, including information about how it is configured.

Keep the **Context Help** window open. It will provide useful information as you complete the rest of this exercise.



Note The Amplitude and Level Measurements Express VI is available only in the LabVIEW Full and Professional Development Systems. If you are using the Base Package, skip the **Analyzing the Amplitude of a Signal** section.

Modifying the Block Diagram

The Simulate Signal Express VI simulates a sine wave by default. You can customize the simulated signal by changing the options in the **Configure Simulate Signal** dialog box.

Complete the following steps to change the simulated signal from a sine wave to a DC signal with uniform white noise.

1. On the block diagram, right-click the Simulate Signal Express VI and select **Properties** from the shortcut menu to display the **Configure Simulate Signal** dialog box.
2. Select **DC** from the **Signal type** pull-down menu.
3. Place a checkmark in the **Add noise** checkbox to add noise to the DC signal.
4. Enter 0.1 in the **Noise amplitude** text box.

The **Result Preview** section displays a random signal. The **Configure Simulate Signal** dialog box should appear similar to Figure 2-3.

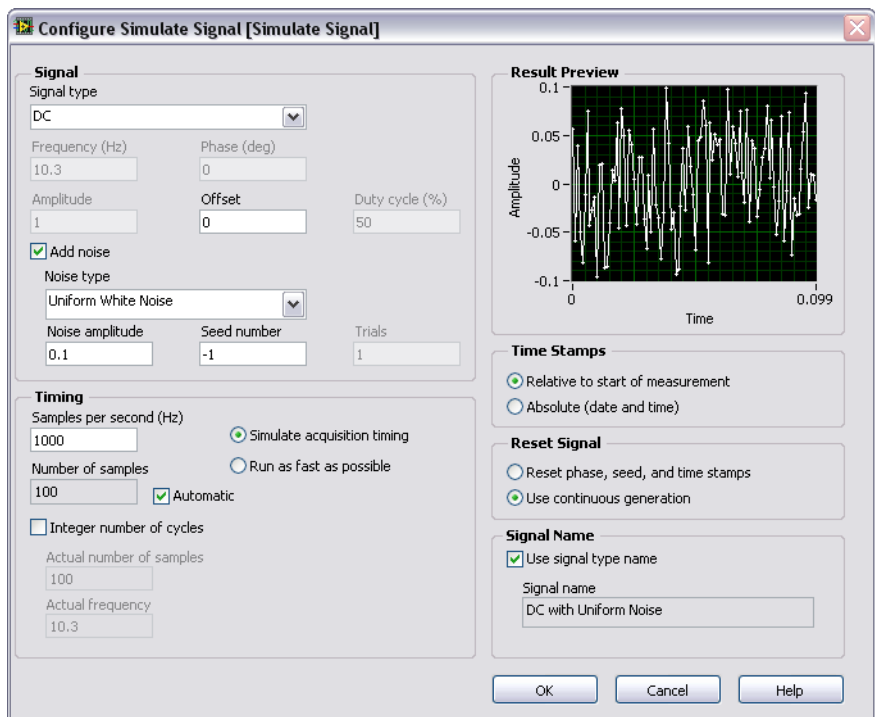


Figure 2-3. Configure Simulate Signal Dialog Box

5. Click the **OK** button to save the current configuration and close the **Configure Simulate Signal** dialog box.
6. Display the front panel by pressing the <Ctrl-E> keys.
7. Run the VI.
The signal appears in the graph and the RMS value for the signal appears in the numeric indicator.
8. Click the **STOP** button to stop the VI.
9. Select **File>Save As** and save the VI as `Analysis.vi` in an easily accessible location.

Modifying the Front Panel

If a template VI contains an indicator you do not want to use, you can delete it.

Complete the following steps to remove the **RMS** indicator from the front panel.

1. On the front panel, move the cursor over the **RMS** indicator until the Positioning tool appears.
2. Click the **RMS** indicator, shown at left, to select it and press the <Delete> key.
3. Display the block diagram.



A wire appears as a dashed black line with a red **x** in the middle, shown at left. This is a broken wire. The **Run** button, shown at left, appears broken to indicate the VI cannot run.

4. Click the broken **Run** button to display the **Error list** window.
The **Error list** window lists all errors in the VI and provides details about each error. You can use the **Error list** window to locate errors. Click the **Help** button for more information about the error.
5. In the **errors and warnings** field, double-click the **Wire: has loose ends** error to highlight the broken wire.
LabVIEW displays the problem causing the error.
6. Press the <Ctrl-B> keys to delete the broken wire.
Pressing the <Ctrl-B> keys deletes *all* broken wires on the block diagram. You can press the <Delete> key to delete only the selected broken wire.

7. Select **View»Error List** to display the **Error list** window. No errors appear in the **errors and warnings** field.



Note You also can press the <Ctrl-L> keys to display the **Error list** window.

8. Click the **Close** button to close the **Error list** window.
The **Run** button no longer appears broken.

Analyzing the Amplitude of a Signal

The Amplitude and Level Measurements Express VI includes options that you can use to analyze the voltage characteristics of a signal.

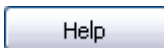
Complete the following steps to reconfigure the Express VI to measure the peak-to-peak amplitude values of the signal.

1. On the block diagram, right-click the Amplitude and Level Measurements Express VI and select **Properties** from the shortcut menu to display the **Configure Amplitude and Level Measurements** dialog box.



Tip You also can double-click the Express VI to display the **Configure Amplitude and Level Measurements** dialog box.

2. In the **Amplitude Measurements** section, remove the checkmark from the **RMS** checkbox.
3. Click the **Help** button, shown at left, in the bottom right corner of the **Configure Amplitude and Level Measurements** dialog box to display the *LabVIEW Help* topic for this Express VI.



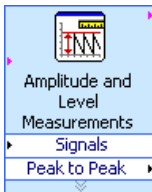
You also can click the **Detailed help** link in the **Context Help** window to display the *LabVIEW Help* topic for this Express VI.

The *Amplitude and Level Measurements* help topic describes the Express VI, the inputs and outputs of the Express VI, and the configuration dialog box options. Each Express VI has a corresponding help topic you can access by clicking the **Help** button.

4. In the *Amplitude and Level Measurements* topic, find the output parameter whose description indicates that it takes a measurement from the most positive peak to the most negative peak of the signal.
5. Minimize the *LabVIEW Help* to return to the **Configure Amplitude and Level Measurements** dialog box.

6. Select the input or output you decided to use.

The option you selected, **Peak to peak**, appears in the **Results** section with the corresponding value of the measurement.



7. Click the **OK** button to save the current configuration and close the **Configure Amplitude and Level Measurements** dialog box.

The **RMS** output of the Amplitude and Level Measurements Express VI changes to reflect the new **Peak to Peak** parameter, shown at left.

Adding a Warning Light

If you want a visual cue indicating when a value exceeds a specified limit, use a warning light.

Complete the following steps to add a warning light to the VI.

1. On the front panel, display the **Controls** palette by selecting **View» Controls Palette**.



Note You also can right-click any blank space on the front panel or the block diagram to display the **Controls** or **Functions** palettes.

2. On the **Express** palette, select the **LEDs** palette, shown in Figure 2-4.

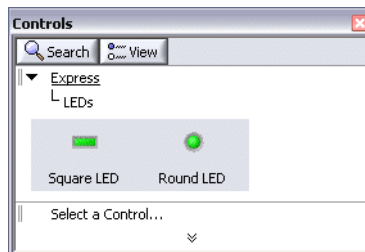


Figure 2-4. LEDs Palette

3. Select the round LED indicator and place it on the front panel to the left of the waveform graph.
4. Double-click the **Boolean** label above the LED and enter `Warning` to change the label of the LED. You also can change the label on the **Appearance** page of the **Boolean Properties** dialog box.

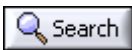
You will use this LED in a later exercise to indicate when a value has exceeded its limit.

5. Select **File»Save As** to display the **Save As** dialog box.
6. Read the various dialog box options. Select the **Copy** and **Substitute copy for original** radio buttons to create a copy of the original VI and immediately edit the copy.
7. Click the **Continue** button and save the VI as `Warning Light.vi` in an easily accessible location.

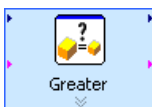
Setting a Warning Level Limit

To specify the value at which you want the warning light to turn on, use the Comparison Express VI.

Complete the following steps to compare the peak-to-peak value to a limit you set.



1. On the block diagram, display the **Functions** palette by selecting **View»Functions Palette**.
2. Click the **Search** button, shown at left, on the **Functions** palette, and enter `Comparison` in the text box. LabVIEW searches as you type and displays any matches in the search results text box. LabVIEW displays a folder glyph to the left of subpalettes in the search results and displays a light blue glyph to the left of Express VIs in the search results.
3. Double-click **Comparison <<Express Comparison>>** to display the **Express Comparison** subpalette and temporarily highlight the Comparison Express VI on the subpalette.
4. Select the Comparison Express VI on the **Express Comparison** palette and place it to the right of the Amplitude and Level Measurements Express VI. The **Configure Comparison** dialog box appears.
5. In the **Compare Condition** section, select the **> Greater** option.
6. In the **Comparison Inputs** section, select **Value** and enter `0.195` in the **Value** text box to assign a constant value at which you want the warning light to turn on.
7. Click the **OK** button to save the current configuration and close the **Configure Comparison** dialog box.



The name of the Comparison Express VI changes to reflect the operation of the Express VI, shown at left. **Greater** indicates that the Express VI does a greater than comparison.

8. Wire the **Peak to Peak** output of the Amplitude and Level Measurements Express VI to the **Operand 1** input of the Greater Express VI.

9. Move the cursor over the wire that connects the **Peak to Peak** output to the **Operand 1** input.
10. When the Positioning tool appears, right-click the wire that connects the **Peak to Peak** output to the **Operand 1** input and select **Create» Numeric Indicator** from the shortcut menu.



A **Peak to Peak** terminal, shown at left, appears on the block diagram. If the **Peak to Peak** terminal appears to be on top of the wires between the Express VIs, move the Express VIs and **Peak to Peak** terminal around to create more space. For example, move the **Peak to Peak** terminal into blank space above the Express VIs.

Warning the User

After you specify the values at which you want the warning light to turn on, you must wire the warning light to the Greater Express VI.

Complete the following steps to provide a visual cue when the peak-to-peak value of the signal exceeds a specified limit.

1. On the block diagram, move the **Warning** terminal to the right of the Greater Express VI. Make sure the **Warning** terminal is inside the gray loop, as shown in Figure 2-5.

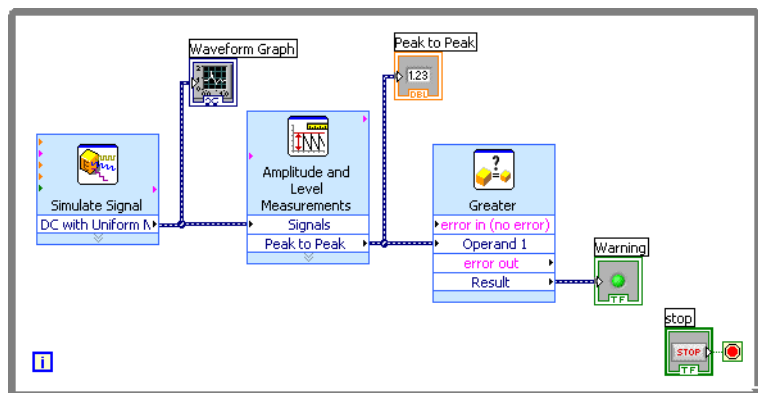


Figure 2-5. Block Diagram of the Warning Light VI

2. Wire the **Result** output of the Greater Express VI to the **Warning** terminal.

The block diagram should appear similar to Figure 2-5.

3. Display the front panel.

A numeric indicator labeled **Peak to Peak** also appears on the front panel. This indicator displays the peak-to-peak value of the signal.

4. Run the VI.

When the peak-to-peak value exceeds 0.195, the **Warning** indicator lights.

5. Click the **STOP** button to stop the VI.
6. Select **File»Save** to save the VI.

Configuring a VI to Save Data to a File

To store information about the data a VI generates, use the Write To Measurement File Express VI.

Complete the following steps to build a VI that saves peak-to-peak values and other information to a LabVIEW data file.

1. On the block diagram, select the Write To Measurement File Express VI from the **Output** palette and place it on the block diagram below and to the right of the Amplitude and Level Measurements Express VI.

The **Configure Write To Measurement File** dialog box opens when you place the Express VI on the block diagram.

The **File Name** text box displays the full path to the output file, `test.lvm`. A `.lvm` file is a tab-delimited text measurement file you can open with a spreadsheet application or a text-editing application. LabVIEW saves data with up to six digits of precision in a `.lvm` file. LabVIEW places the `.lvm` file in the default `LabVIEW Data` directory. LabVIEW installs the `LabVIEW Data` directory in the default file directory of the operating system.

When you want to view the data, use the file path displayed in the **File Name** text box to access the `test.lvm` file.

2. In the **If a file already exists** section of the **Configure Write To Measurement File** dialog box, select the **Append to file** option.
If you select **Append to file**, LabVIEW writes all the data to the `test.lvm` file without erasing any existing data in the file.
3. In the **Segment Headers** section, select the **One header only** option to create only one header in the file to which LabVIEW writes the data.
4. Enter the following text in the **File Description** text box: `Sample of peak to peak values`. LabVIEW appends the text you enter in this text box to the header of the file.
5. Click the **OK** button to save the current configuration and close the **Configure Write To Measurement File** dialog box.

Saving Data to a File

When you run the VI, LabVIEW saves the data to the `test.lvm` file.

Complete the following steps to generate the `test.lvm` file.

1. On the block diagram, wire the **Peak to Peak** output of the Amplitude and Level Measurements Express VI to the **Signals** input of the Write To Measurement File Express VI.
2. Select **File>Save As** and save the VI as `Save Data.vi` in an easily accessible location.
3. Display the front panel and run the VI.
4. Click the **STOP** button on the front panel.
5. To view the data you saved, open the `test.lvm` file with a spreadsheet or text-editing application.

The file has one header, which contains information about the Express VI.

6. Close the file when you finish looking at it and return to the `Save Data VI`.

Adding a Button That Stores Data When Pressed

If you want to store only certain data points, you can configure the Write To Measurement File Express VI to save peak-to-peak values only when a user clicks a button.

Complete the following steps to add a button to the VI and configure how the button responds when a user clicks it.

1. Click the **Search** button on the **Controls** palette, enter `button` in the text box, and select **Buttons & Switches** from the list of controls.
2. Select the rocker button on the **Buttons & Switches** palette and place it to the right of the waveform graph.
3. Right-click the rocker button and select **Properties** from the shortcut menu to display the **Boolean Properties** dialog box.
4. Change the label of the button to `Write to File`.
5. On the **Operation** page of the **Boolean Properties** dialog box, select **Latch when pressed** from the **Button behavior** list.

Use the **Operation** page to specify how a button behaves when a user clicks it. To see how the button reacts to a click, click the button in the **Preview Selected Behavior** section.

6. Click the **OK** button to save the current configuration and close the **Boolean Properties** dialog box.
7. Save the VI.

Saving Data When Prompted by a User

Complete the following steps to build a VI that logs data to a file when the user clicks a button on the front panel.

1. On the block diagram, double-click the Write To Measurement File Express VI to display the **Configure Write To Measurement File** dialog box.
2. Change the filename `test.lvm` in the **File name** text box to `Selected Samples.lvm` to save the data to a different file.
3. Click the **OK** button to save the current configuration and close the **Configure Write To Measurement File** dialog box.
4. Right-click the **Signals** input of the Write To Measurement File Express VI and select **Insert Input/Output** from the shortcut menu to insert the **Comment** input.

In the previous exercise you learned to add inputs and outputs by expanding the Express VI using the down arrows. This method is a different way of displaying and selecting the inputs and outputs of an Express VI.

The inputs and outputs of an Express VI appear in a predetermined order when you add new inputs and outputs. To select a specific input, you might need to add an input first, then change the input to the specific one you want to use by right-clicking the input and selecting **Select Input/Output** from the shortcut menu.

5. Right-click the **Comment** input of the Write To Measurement File Express VI and select **Select Input/Output**►**Enable** from the shortcut menu to insert the **Enable** input.
6. Move the **Write to File** terminal to the left of the Write To Measurement File Express VI.
7. Wire the **Write to File** terminal to the **Enable** input of the Write To Measurement File Express VI.

The block diagram should appear similar to Figure 2-6.

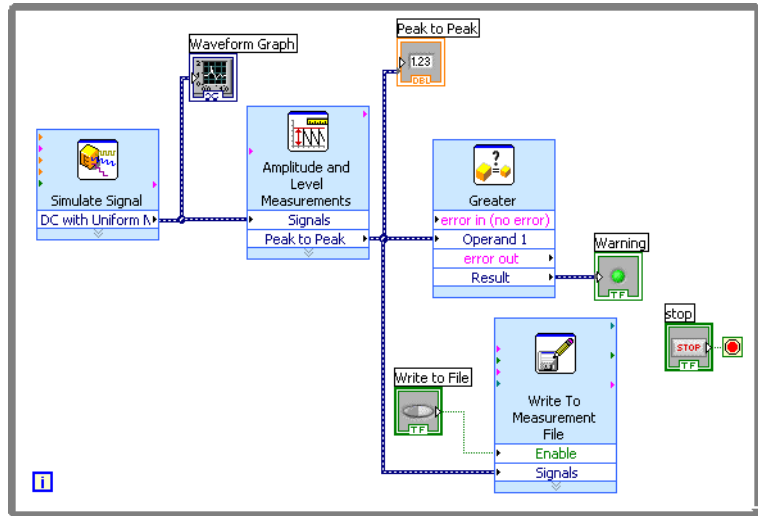


Figure 2-6. Block Diagram of the Save Data VI

Viewing Saved Data

Complete the following steps to view the data that you save to the Selected Samples.lvm file.

1. Display the front panel and run the VI. Click the **Write to File** button several times.
2. Click the **STOP** button on the front panel.
3. Open the Selected Samples.lvm file with a spreadsheet or text-editing application.

The Selected Samples.lvm file differs from the test.lvm file. test.lvm recorded all the data generated by the Save Data VI, whereas Selected Samples.lvm recorded data only when you clicked the **Write to File** button.

4. Save and close the VI.

Summary

The following topics are a summary of the main concepts you learned in this chapter.

LabVIEW Documentation Resources

LabVIEW includes extensive documentation for new and experienced LabVIEW users. Documentation resources include the **Context Help** window, the *LabVIEW Help*, examples, and manuals.

The **Context Help** window displays basic information about LabVIEW objects when you move the cursor over each object. To access the **Context Help** window, select **Help»Show Context Help** or press the <Ctrl-H> keys.

The *LabVIEW Help* contains information about LabVIEW programming concepts, step-by-step instructions for using LabVIEW, and reference information about LabVIEW VIs, functions, palettes, menus, and tools. To access help information for Express VIs, click the **Help** button, shown at left, in the configuration dialog box while configuring an Express VI. You also can access the *LabVIEW Help* by clicking the **Detailed help** link in the **Context Help** window, by right-clicking a VI or function on the block diagram or on a pinned palette and selecting **Help** from the shortcut menu, or by selecting **Help»Search the LabVIEW Help**.



Controls and Indicators

You can configure the controls and indicators on the front panel to perform tasks depending on what you want a VI to do. In this chapter, you learned to use controls and indicators in the following ways:

- You can build VIs that perform a task when certain conditions occur, such as displaying a warning light when a value exceeds a certain limit.
- You can build VIs that let users control when an Express VI executes by using buttons and the **Enable** input. You can configure the buttons to operate in one of six ways using the **Operation** page of the **Boolean Properties** dialog box.

Saving Data

The Write To Measurement File Express VI saves data that a VI generates and analyzes to a `.lvmm` or `.tdm` measurement file. The text-based measurement file (`.lvmm`) is a tab-delimited text file you can open with a spreadsheet application or a text-editing application. LabVIEW saves data

with up to six digits of precision in a `.lvnm` file. In addition to the data an Express VI generates, the `.lvnm` file includes headers that contain information about the data, such as the date and time LabVIEW generated the data. The binary measurement file (`.tdm`) is a binary file that contains waveform data. Binary `.tdm` files provide higher accuracy for floating-point numbers, have a smaller disk footprint, and perform faster than text-based measurement files (`.lvnm`).

LabVIEW installs the `LabVIEW Data` directory in the default file directory of the operating system to help you organize and locate the data files LabVIEW generates. Refer to the *LabVIEW Help* for more information about saving data to and retrieving data from `.lvnm` and `.tdm` files.

Errors and Broken Wires

The **Run** button appears broken when the VI you are creating or editing contains errors. If the **Run** button is still broken when you finish wiring the block diagram, the VI is broken and cannot run.

Click the broken **Run** button or select **View»Error List** to find out why a VI is broken. You can use the **Error list** window to locate errors. Click the **Help** button for more information about the error. Double-click the error in the **errors and warnings** field to highlight the problem causing the error.

A broken wire appears as a dashed black line with a red **x** in the middle. Broken wires occur for a variety of reasons, such as if you delete wired objects. The VI cannot run if the block diagram contains broken wires.

Move the Wiring tool over a broken wire to display a tip strip that describes why the wire is broken. This information also appears in the **Context Help** window when you move the Wiring tool over a broken wire. Right-click the wire and select **List Errors** from the shortcut menu to display the **Error list** window. Click the **Help** button for more information about why the wire is broken.

Customizing a VI

You can choose one of many LabVIEW template VIs to use as a starting point when building VIs. However, sometimes you need to build a VI for which a template is not available. This chapter teaches you how to build and customize a VI without using a template.

Building a VI from a Blank VI

In the following exercises, you will open a blank VI and add Express VIs and structures to the block diagram to build a new VI. You will build a VI that generates a signal, reduces the number of samples in the signal, and displays the resulting data in a table on the front panel. When you complete the exercises, the front panel of the VI will look similar to the front panel in Figure 3-1.



You can complete the exercises in this chapter in approximately 30 minutes.

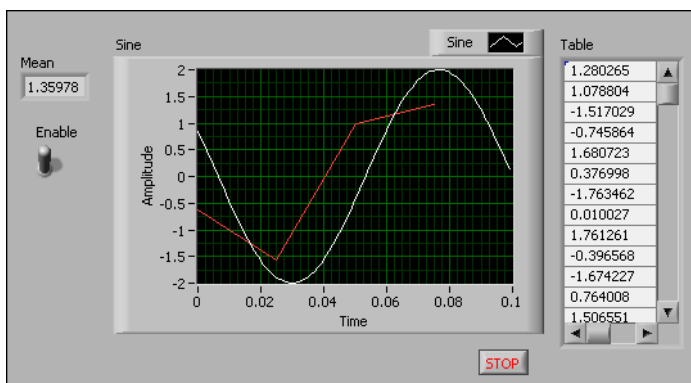


Figure 3-1. Front Panel of the Reduce Samples VI

Opening a Blank VI

If no template is available for the VI you want to build, you can start with a blank VI and add Express VIs to accomplish a specific task.

Complete the following steps to open a blank VI.

1. In the **Getting Started** window, click the **Blank VI** link in the **New** section or press the <Ctrl-N> keys to open a blank VI.

A blank front panel and block diagram appear.



Note You also can open a blank VI by selecting **File»New VI** or by selecting **File»New** and selecting **Blank VI** from the **Create New** list.

2. If the **Functions** palette is not visible, right-click any blank space on the block diagram to display a temporary version of the **Functions** palette. Click the thumbtack, shown at left, in the upper left corner of the **Functions** palette to pin the palette so it is no longer temporary.



Adding an Express VI That Simulates a Signal

Complete the following steps to find the Express VI you want to use and add it to the block diagram.

1. If the **Context Help** window is not visible, press the <Ctrl-H> keys to display the window. You also can click the **Show Context Help Window** button, shown at left, on the front panel or block diagram toolbar to display the **Context Help** window.



2. On the **Functions** palette, select the **Express»Input** palette and move the cursor over the Express VIs on the **Input** palette.

The **Context Help** window displays information about the behavior of each Express VI.

3. Use the information that appears in the **Context Help** window to find the Express VI that can generate a sine wave signal.
4. Select the Express VI and place it on the block diagram. The **Configure Simulate Signal** dialog box appears.
5. Move the cursor over the various options in the **Configure Simulate Signal** dialog box, such as **Frequency (Hz)**, **Amplitude**, and **Samples per second (Hz)**. Read the information that appears in the **Context Help** window.

6. Configure the Simulate Signal Express VI to generate a sine wave with a frequency of 10.7 and amplitude of 2.

The signal in the **Result Preview** window changes to reflect the configured sine wave.

7. Click the **OK** button to save the current configuration and close the **Configure Simulate Signal** dialog box.

8. Move the cursor over the Simulate Signal Express VI and read the information that appears in the **Context Help** window.
The **Context Help** window now displays the configuration information of the Simulate Signal Express VI.
9. Save the VI as `Reduce Samples.vi` in an easily accessible location.

Searching the Help and Modifying a Signal

Complete the following steps to use the *LabVIEW Help* to search for the Express VI that reduces the number of samples in a signal.

1. Move the cursor over the Simulate Signal Express VI and click the **Detailed help** link in the **Context Help** window to display the *Simulate Signal* topic in the *LabVIEW Help*. You might have to enlarge or scroll down in the **Context Help** window to see the **Detailed help** link.

You also can access the *LabVIEW Help* by right-clicking a VI or function on the block diagram or on a pinned palette and selecting **Help** from the shortcut menu or by selecting **Help»Search the LabVIEW Help**.

2. Click the **Search** tab, enter `sample compression` in the **Type in the word(s) to search for** text box, and press the <Enter> key. You can place quotation marks around the phrase to search for the exact phrase. For example, you can enter "sample compression" to narrow the search results.

This word choice reflects what you want this Express VI to do—compress, or reduce, the number of samples in a signal.

3. Double-click the *Sample Compression* topic to display the topic that describes the Sample Compression Express VI.
4. After you read the description of the Express VI, click the square gray **Place on the block diagram** button to place the Express VI on the cursor.
5. Move the cursor to the block diagram.
6. Place the Sample Compression Express VI on the block diagram to the right of the Simulate Signal Express VI.
7. Configure the Sample Compression Express VI to reduce the signal by a factor of 25 using the mean of these values.
8. Click the **OK** button to save the current configuration and close the **Configure Sample Compression** dialog box.

- Use the Wiring tool to wire the **Sine** output of the Simulate Signal Express VI to the **Signals** input of the Sample Compression Express VI.

Customizing a User Interface from the Block Diagram

In the previous exercises, you added controls and indicators to the front panel using the **Controls** palette. You also can create controls and indicators from the block diagram.

Complete the following steps to create controls and indicators from the block diagram.

- On the block diagram, right-click the **Mean** output of the Sample Compression Express VI and select **Create»Numeric Indicator** from the shortcut menu to create a numeric indicator.
- Right-click the **Mean** output of the Sample Compression Express VI and select **Insert Input/Output** from the shortcut menu to insert the **Enable** input.
- Right-click the **Enable** input and select **Create»Control** from the shortcut menu to create the **Enable** switch.

Control terminals have a thicker border than indicator terminals. Also, an arrow appears on the right of the terminal if the terminal is a control, and an arrow appears on the left of the terminal if the terminal is an indicator.

- Right-click the wire that connects the **Sine** output of the Simulate Signal Express VI to the **Signals** input of the Sample Compression Express VI and select **Create»Graph Indicator** from the shortcut menu.
- Use the Wiring tool to wire the **Mean** output of the Sample Compression Express VI to the **Sine** graph indicator.
The Merge Signals function appears.
- Arrange the objects on the block diagram so they appear similar to Figure 3-2.



Tip You can right-click any wire and select **Clean Up Wire** from the shortcut menu to have LabVIEW automatically find a route for the wire around existing objects on the block diagram. LabVIEW also routes a wire to decrease the number of bends in the wire.

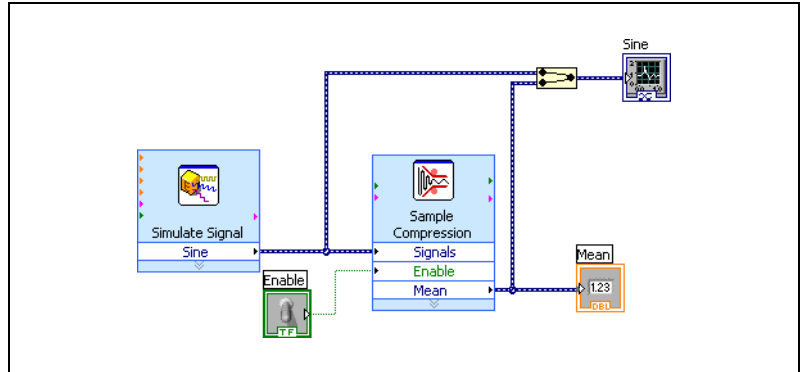


Figure 3-2. Block Diagram of the Reduce Samples VI

7. Display the front panel.

The controls and indicators you added appear on the front panel with labels that correspond to the inputs and outputs from which you created the controls and indicators.

8. Save the VI.

Configuring a VI to Run Continuously until the User Stops It

In the current state, the VI runs once, generates one signal, then stops executing. To run the VI until a condition occurs, you can use a While Loop.

Complete the following steps to add a While Loop to the block diagram.

1. Display the front panel and run the VI.
The VI runs once and then stops. The front panel does not have a stop button.
2. On the block diagram, select the While Loop on the **Execution Control** palette.
3. Move the cursor to the upper left corner of the block diagram. Click to place the top left corner of the While Loop.

4. Drag the cursor diagonally to enclose *all* the Express VIs and wires, as shown in Figure 3-3.

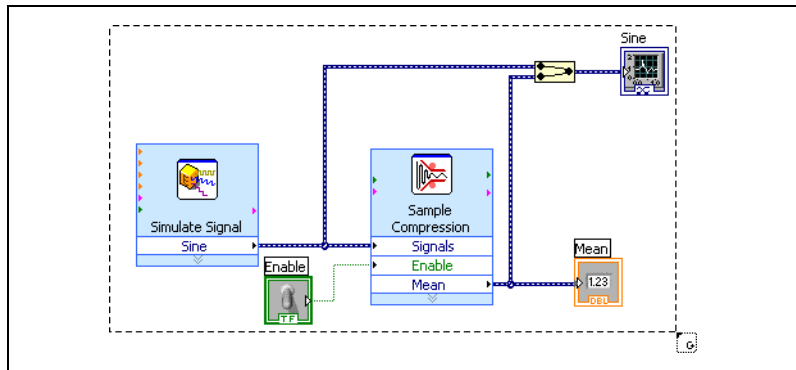
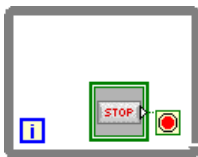


Figure 3-3. Placing the While Loop around the Express VIs



The While Loop, shown at left, appears with a **STOP** button wired to the conditional terminal. This While Loop is configured to stop when the user clicks the **STOP** button.

5. Display the front panel and run the VI.

The VI now runs until you click the **STOP** button. A While Loop executes the VIs and functions inside the loop until the user clicks the **STOP** button.

6. Click the **STOP** button and save the VI.

Searching for Examples

To learn more about how you can use a certain VI, you can search for and view an example that uses the VI.

Complete the following steps to find and open an example that uses the Time Delay Express VI.

1. Select **Help»Search the LabVIEW Help** to display the *LabVIEW Help*.
2. Click the **Search** tab, enter "time delay" in the **Type in the word(s) to search for** text box, and press the <Enter> key.

Before you search, you can narrow the search results by placing a checkmark in the **Search titles only** checkbox near the bottom of the help window. You also can use operators such as AND, OR, and NEAR in the **Type in the word(s) to search for** text box to narrow the search results.

3. Click the **Location** column heading to sort the search results by content type. *Reference* topics contain reference information about block diagram objects such as VIs, functions, properties, and methods. *How-To* topics contain step-by-step instructions for using LabVIEW. *Concept* topics contain information about LabVIEW programming concepts.
4. Double-click the *Time Delay* topic to display the topic that describes the Time Delay Express VI.
5. After you read the description of the Express VI, click the **Open example** button in the *Example* section near the bottom of the topic to open an example that uses the Time Delay Express VI.
6. Click the **Browse related examples** button to open the NI Example Finder and display a list of other examples that use this VI. The NI Example Finder searches among hundreds of example VIs, including all installed examples and the examples located on NI Developer Zone at ni.com/zone. You can modify an example VI to fit an application, or you can copy and paste from one or more examples into a VI that you create.

You also can select **Help»Find Examples** or click the **Find Examples** link in the **Examples** section of the **Getting Started** window to launch the NI Example Finder. You can right-click a VI or function on the block diagram or on a pinned palette and select **Examples** from the shortcut menu to display a help topic with links to examples for that VI or function.

7. After you experiment with the NI Example Finder and the example VIs, close the NI Example Finder and the example VIs to return to the Reduce Samples VI.

Controlling the Speed of Execution

To plot the points on the waveform graph more slowly, you can add a time delay to the block diagram.

Complete the following steps to control the speed at which the VI runs.

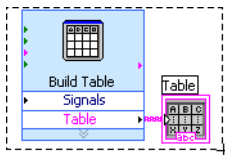
1. On the block diagram, search for the Time Delay Express VI on the **Functions** palette and place it inside the While Loop.
2. Enter 0.250 in the **Time delay (seconds)** text box.
This time delay specifies how fast the loop runs. With a 0.250 second time delay, the loop iterates once every quarter of a second.
3. Click the **OK** button to save the current configuration and close the **Configure Time Delay** dialog box.

4. Display the front panel and run the VI.
5. Click the **Enable** switch and examine the change on the graph.
If the **Enable** switch is on, the graph displays the reduced signal. If the **Enable** switch is off, the graph does not display the reduced signal.
6. Click the **STOP** button to stop the VI.

Using a Table to Display Data

Complete the following steps to display a collection of mean values in a table on the front panel.

1. On the front panel, search for the **Express Table** indicator on the **Controls** palette and place it on the front panel to the right of the waveform graph.
2. Display the block diagram.
LabVIEW wired the **Table** terminal to the Build Table Express VI.
3. If the Build Table Express VI and the **Table** terminal are not selected already, click an open area on the block diagram to the left of the Build Table Express VI and the **Table** terminal. Drag the cursor diagonally until the selection rectangle encloses the Build Table Express VI and the **Table** terminal, shown at left.



A moving dashed outline, called a marquee, highlights the Build Table Express VI, the **Table** terminal, and the wire joining the two.

4. Drag the objects into the While Loop to the right of the **Mean** terminal.
The While Loop resizes to enclose the Build Table Express VI and the **Table** terminal.
5. Use the Wiring tool to wire the **Mean** terminal of the Sample Compression Express VI to the **Signals** input of the Build Table Express VI.

The block diagram should appear similar to Figure 3-4.

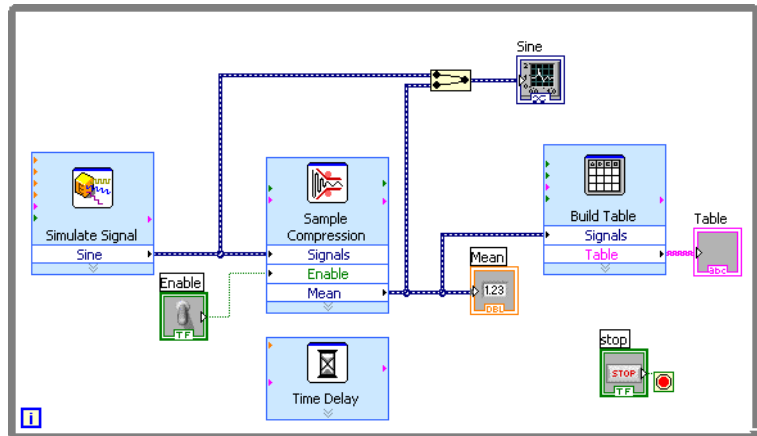


Figure 3-4. Block Diagram of the Reduce Samples VI

6. Display the front panel and run the VI.
7. Click the **Enable** switch.
The table displays the mean values of every 25 samples of the sine wave. If the **Enable** switch is off, the table does not record the mean values.
8. Stop the VI.
9. Experiment with properties of the table by using the **Table Properties** dialog box. For example, try changing the number of columns to one.
10. Save and close the VI.

Summary

The following topics are a summary of the main concepts you learned in this chapter.

Using the LabVIEW Help Resources

You can use the **Context Help** window and the *LabVIEW Help* to learn more about Express VIs. Both provide information that describe the functionality of the Express VI and how to configure the Express VI.

In this chapter, you learned to use the help resources in the following ways:

- The **Context Help** window displays basic information about LabVIEW objects when you move the cursor over each object. Objects with context help information include VIs, structures, palettes, dialog box components, and so on.
- When you move the cursor over an Express VI on the block diagram, the **Context Help** window displays a brief description of the Express VI and information about how you configured the Express VI.
- If you find an Express VI or other block diagram object in the *LabVIEW Help* you want to use, you can click a **Place on the block diagram** button to place the object on the block diagram.
- To navigate the *LabVIEW Help*, use the **Contents**, **Index**, and **Search** tabs. Use the **Contents** tab to get an overview of the topics and structure of the help. Use the **Index** tab to find a topic by keyword. Use the **Search** tab to search the help for a word or phrase.
- On the **Search** tab of the *LabVIEW Help*, use operators such as AND, OR, and NEAR to narrow the search results. To search for an exact phrase, place quotation marks around the phrase. Before you search, you also can narrow the search results by placing a checkmark in the **Search titles only** checkbox near the bottom of the help window.
- On the **Search** tab of the *LabVIEW Help*, you can click the **Location** column heading above the list of search results to sort the results by content type. *Reference* topics contain reference information about block diagram objects such as VIs, functions, properties, and methods. *How-To* topics contain step-by-step instructions for using LabVIEW. *Concept* topics contain information about LabVIEW programming concepts.

Customizing the Block Diagram Code

You can use many controls, indicators, Express VIs, and structures to build a VI. To customize a VI, you can create controls and indicators, control when a VI stops running, and display generated data in a table.

Creating Controls and Indicators

Create controls and indicators on the block diagram by right-clicking the Express VI input, output, or wire, selecting **Create** from the shortcut menu, and selecting among the available options. LabVIEW wires the control or indicator you created to the input, output, or wire you right-clicked.

Control terminals have a thicker border than indicator terminals. Also, an arrow appears on the right of the terminal if the terminal is a control, and an arrow appears on the left of the terminal if the terminal is an indicator.

Controlling When a VI Stops Running

Use a While Loop to run the code enclosed within the loop continually. A While Loop stops running when a stop condition occurs. When you place or move an object in a While Loop near the border, the loop resizes to add space for that object.

The **Execution Control** palette includes objects you can use to control the number of times a VI runs, as well as the speed at which the VI runs.

Displaying Data in a Table

The table indicator displays generated data. Use the Build Table Express VI to build a table of generated data.

Optional: Acquiring Data and Communicating with Instruments

This chapter introduces you to the Express VIs you use to acquire data and communicate with instruments on Windows. These exercises require data acquisition hardware.

Refer to the **Taking Measurements** book on the **Contents** tab in the *LabVIEW Help* for information about acquiring data and communicating with instruments on all platforms.

Acquiring a Signal

In the following exercises, you will use the DAQ Assistant Express VI to create a task in NI-DAQmx. NI-DAQmx is a programming interface you can use to communicate with data acquisition devices. Refer to the **Getting Started»Taking an NI-DAQmx Measurement in LabVIEW** book on the **Contents** tab in the *LabVIEW Help* for information about additional methods you can use to create NI-DAQmx tasks.



Note The following exercises require that you have NI-DAQmx and an NI-DAQmx-supported device. Refer to the National Instruments Web site at ni.com/daq for the list of NI-DAQmx-supported devices. If you do not have NI-DAQmx or an NI-DAQmx-supported device, refer to the **Taking Measurements** book on the **Contents** tab in the *LabVIEW Help* for information about using Traditional NI-DAQ (Legacy) for data acquisition.

In the following exercises, you will create an NI-DAQmx task that continuously takes a voltage reading and plots the data on a waveform graph.



You can complete the exercises in this chapter in approximately 30 minutes.

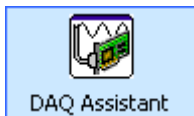
Creating an NI-DAQmx Task

In NI-DAQmx, a task is a collection of one or more channels, timing, triggering, and other properties. Conceptually, a task represents a measurement or generation you want to perform. For example, you can create a task to measure temperature from one or more channels on a DAQ device.

Complete the following steps to create and configure a task that reads a voltage level from a DAQ device.

1. Open a new VI.
2. On the block diagram, select the **Input** palette on the **Functions** palette.

If the **Functions** palette is not visible, select **View»Functions Palette**.



3. Select the DAQ Assistant Express VI, shown at left, on the **Input** palette and place it on the block diagram. The DAQ Assistant launches and the **Create New** dialog box appears.
4. Click **Analog Input** to display the **Analog Input** options.
5. Select **Voltage** to create a new voltage analog input task.

The dialog box displays a list of channels on each installed DAQ device. The number of channels listed depends on the number of channels you have on the DAQ device.

6. In the **Supported Physical Channels** list, select the physical channel to which the instrument connects the signal, such as **ai0**, and then click the **Finish** button. The DAQ Assistant opens a new dialog box, shown in Figure 4-1, which displays options for configuring the channel you selected to complete a task.

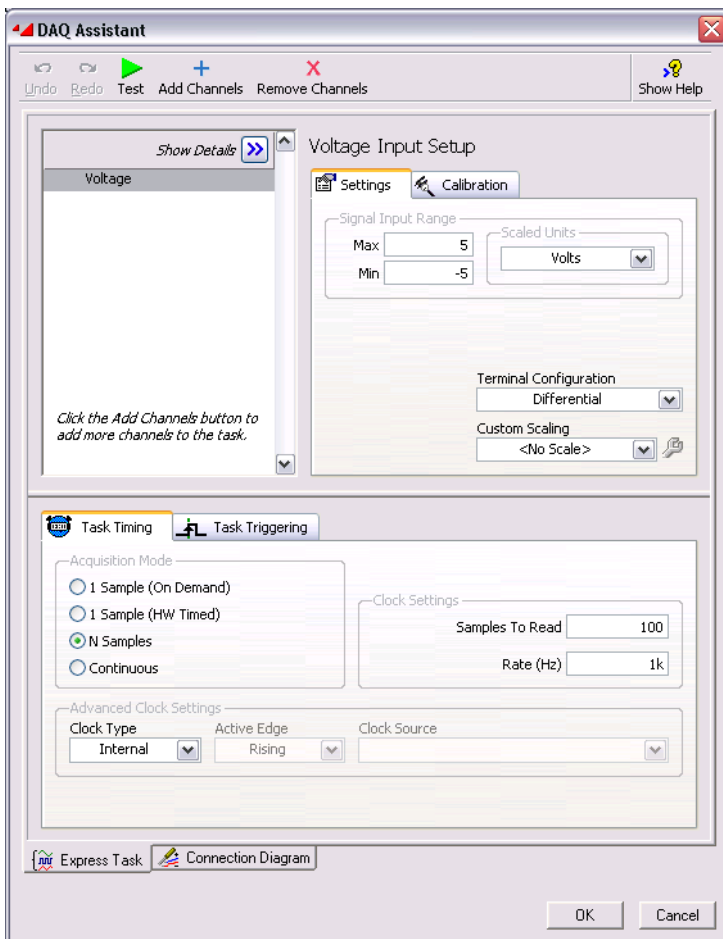


Figure 4-1. Configuring a Task Using the DAQ Assistant

7. In the **Input Range** section of the **Settings** page, enter 10 for the **Max** value and enter -10 for the **Min** value.
8. On the **Task Timing** page, select the **N Samples** option.
9. Enter a value of 1000 in the **Samples To Read** text box.

Testing the Task



You can test the task to verify that you correctly configured the channel.

Complete the following steps to confirm that you are acquiring data.

1. Click the **Test** button, shown at left. The **DAQ Assistant** dialog box appears.
2. Click the **Start** button once or twice to confirm that you are acquiring data, and then click the **Close** button to return to the DAQ Assistant.
3. Click the **OK** button to save the current configuration and close the DAQ Assistant.
4. Save the VI as `Read Voltage.vi` in an easily accessible location.

Graphing Data from a DAQ Device

You can use the task you created in the previous exercise to graph the data acquired from a DAQ device.

Complete the following steps to plot the data from the channel on a waveform graph and change the name of the signal.

1. On the block diagram, right-click the **data** output and select **Create» Graph Indicator** from the shortcut menu.
2. Display the front panel and run the VI three or four times. Observe the waveform graph.
voltage appears in the waveform graph plot legend.
3. On the block diagram, right-click the DAQ Assistant Express VI and select **Properties** from the shortcut menu to open the DAQ Assistant.
4. Right-click **Voltage** in the list of channels and select **Rename** from the shortcut menu to display the **Rename a channel or channels** dialog box.



Tip You also can select the name of the channel and press the <F2> key to display the **Rename a channel or channels** dialog box.

5. In the **New Name** text box, enter `First Voltage Reading`, and click the **OK** button.
6. Click the **OK** button to save the current configuration and close the DAQ Assistant.
7. Display the front panel and run the VI.
First Voltage Reading appears in the waveform graph plot legend.
8. Save the VI.

Editing an NI-DAQmx Task

You can add a channel to the task so you can compare two separate voltage readings. You also can customize the task to acquire the voltage readings continuously.

Complete the following steps to add a new channel to the task and acquire data continuously.



1. On the block diagram, double-click the DAQ Assistant Express VI to open the DAQ Assistant.
2. Click the **Add Channels** button, shown at left, and select the Voltage channel from the **Add Channel** menu to display the **Add Channels To Task** dialog box.
3. Select any unused physical channel in the **Supported Physical Channels** list, and click the **OK** button to return to the DAQ Assistant.
4. Rename the channel `Second Voltage Reading`.
5. On the **Task Timing** page, select the **Continuous** option.

When you set timing and triggering options in the DAQ Assistant, these options apply to all the channels in the list of channels.

6. Click the **OK** button to save the current configuration and close the DAQ Assistant. The **Confirm Auto Loop Creation** dialog box appears.
7. Click the **Yes** button. LabVIEW places a While Loop around the DAQ Assistant Express VI and the graph indicator. A stop button appears on the block diagram wired to the **stop** input of the DAQ Assistant Express VI. The **stopped** output of the Express VI is wired to the conditional terminal of the While Loop. The block diagram should appear similar to Figure 4-2.

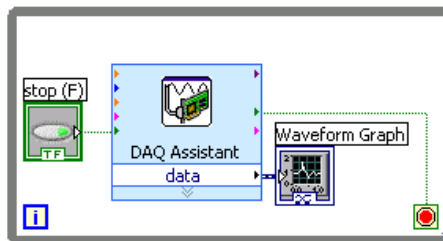


Figure 4-2. Block Diagram of the Read Voltage VI

If an error occurs or you click the **stop** button while the VI is running, the DAQ Assistant Express VI stops reading data and the **stopped** output returns a TRUE value and stops the While Loop.

Visually Comparing Two Voltage Readings

Because you have two voltage readings displayed on a graph, you can customize the plots to distinguish between the two.

Complete the following steps to customize the plot color on the waveform graph.

1. On the front panel, expand the plot legend to display two plots.
2. Run the VI.

Two plots appear on the graph, and the plot legend displays both plot names.

3. Right-click **First Voltage Reading** in the plot legend and select **Color** from the shortcut menu. Using the color picker, select a color such as yellow so the plot is easy to read. Change the plot color of **Second Voltage Reading**.
4. Save the VI.

Communicating with an Instrument

Instrument drivers simplify instrument control and reduce test program development time by eliminating the need to learn the programming protocol for each instrument. An instrument driver is a set of software routines that control a programmable instrument. Each routine corresponds to a programmatic operation such as configuring, reading from, writing to, and triggering the instrument. Use an instrument driver for instrument control when possible. National Instruments provides thousands of instrument drivers for a wide variety of instruments.

Use the NI Instrument Driver Finder to search for and install instrument drivers without leaving the LabVIEW development environment. Select **Help»Find Instrument Drivers** to launch the Instrument Driver Finder. You also can visit the NI Instrument Driver Network at ni.com/idnet to find a driver for an instrument.

If a driver is not available for an instrument, you can use the Instrument I/O Assistant Express VI to communicate with the instrument.



Note You must have the Instrument I/O Assistant and an instrument installed to use the Instrument I/O Assistant Express VI. You install the Instrument I/O Assistant from the National Instruments Device Drivers CD.

In the following exercises, you will communicate with an instrument.

Selecting an Instrument

Before you communicate with an instrument, you must select the instrument with which you want to communicate.

Complete the following steps to use the Instrument I/O Assistant Express VI to select an instrument.

1. Turn on the instrument you want to use. The instrument must be powered on to use the Instrument I/O Assistant Express VI.
2. Select the Instrument I/O Assistant Express VI on the **Input** palette and place it on the block diagram. The **Instrument I/O Assistant** dialog box appears.
3. Click the **Show Help** button, shown at left, in the upper right corner of the **Instrument I/O Assistant** dialog box.



The help appears to the right of the dialog box. The top help window contains how-to information about using the Instrument I/O Assistant. The bottom help window provides context-sensitive help about components in the dialog box.

4. Click the **Select Instrument** link in the top help window and follow the instructions in the help window to select the instrument with which you want to communicate.
5. If necessary, configure the properties of the instrument.
6. If you want to minimize the help window, click the **Hide Help** button, shown at left, in the upper right corner of the **Instrument I/O Assistant** dialog box.



Acquiring and Parsing Information for an Instrument

After you select the instrument, you can send commands to the instrument to retrieve data. In this exercise, you will learn to use the Instrument I/O Assistant Express VI to acquire and parse identification information for an instrument.

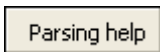
Complete the following steps to communicate with the instrument.

1. In the **Instrument I/O Assistant** dialog box, click the **Add Step** button and click the **Query and Parse** step.

2. Enter *IDN? in the **Enter a command** text box.
*IDN? is a query that most instruments recognize. The response is an identification number string that describes the instrument. If the instrument does not accept this command, refer to the reference manual for the instrument for a list of commands the instrument does accept.



3. Click the **Run** button, shown at left.
The Instrument I/O Assistant sends the command to the instrument, and the instrument returns its identification information.
4. Select **ASCII only** from the pull-down menu below the **Byte index** column of the response window to parse the instrument name as an ASCII string. You also can use the Instrument I/O Assistant to parse ASCII numbers and binary data.



5. Click the **Parsing help** button, shown at left, in the **Instrument I/O Assistant** dialog box to display information about parsing data.
6. In the **ASCII representation** column of the response window, click the value you want to parse.
7. Enter a name for the token, or parsed data selection, in the **Token name** text box.
8. Click the **OK** button to save the current configuration and close the **Instrument I/O Assistant** dialog box.

The name that you entered in the **Token name** text box is the output of the Instrument I/O Assistant Express VI, shown at left.



Summary

The following topics are a summary of the main concepts you learned in this chapter.

DAQ Assistant Express VI

You can use the DAQ Assistant Express VI to interactively build measurement channels or tasks.

Place the DAQ Assistant Express VI on the block diagram to configure channels and tasks for use with NI-DAQmx for data acquisition.

NI-DAQmx is a programming interface you can use to communicate with data acquisition devices. You can use the DAQ Assistant Express VI to control devices supported by NI-DAQmx.

Refer to the **Getting Started»Taking an NI-DAQmx Measurement in LabVIEW** book on the **Contents** tab in the *LabVIEW Help* for information about the DAQ Assistant.

Refer to the National Instruments Web site at ni.com/daq for information about devices supported by NI-DAQmx. If NI-DAQmx does not support the device you want to use, refer to the **Taking Measurements** book on the **Contents** tab in the *LabVIEW Help* for information about using Traditional NI-DAQ (Legacy) for data acquisition.

Tasks

In NI-DAQmx, a task is a collection of one or more channels, timing, triggering, and other properties. Conceptually, a task represents a measurement or generation you want to perform.

For example, you can configure a collection of channels for analog input operations. After you create a task, you can access the single task instead of configuring the channels individually to perform analog input operations. After you create a task, you can add or remove channels from that task.

Refer to the **Taking Measurements** book on the **Contents** tab in the *LabVIEW Help* for more information about channels and tasks.

Instrument I/O Assistant Express VI

An instrument driver is a set of software routines that control a programmable instrument. Each routine corresponds to a programmatic operation such as configuring, reading from, writing to, and triggering the instrument. Use an instrument driver for instrument control when possible. National Instruments provides thousands of instrument drivers for a wide variety of instruments.

Use the NI Instrument Driver Finder to search for and install instrument drivers without leaving the LabVIEW development environment. Select **Help»Find Instrument Drivers** to launch the Instrument Driver Finder. You also can visit the NI Instrument Driver Network at ni.com/idnet to find a driver for an instrument.

If a driver is not available for an instrument, you can use the Instrument I/O Assistant Express VI to communicate with the instrument. You can use the Instrument I/O Assistant to communicate with message-based instruments and graphically parse the response. Start the Instrument I/O Assistant by placing the Instrument I/O Assistant Express VI on the block diagram or by double-clicking the Instrument I/O Assistant Express VI icon on the block diagram.

Refer to the *Instrument I/O Assistant Help* for information about communicating with an external device. Display the *Instrument I/O Assistant Help* by clicking the **Show Help** button in the **Instrument I/O Assistant** dialog box.

Using Other LabVIEW Features

The previous chapters in this manual introduce you to most of the LabVIEW features you need to build common measurement applications. As you familiarize yourself with the LabVIEW environment, you might find that you need to enhance VIs or that you need more fine-tuned control of the processes the VIs perform. This chapter introduces you to some of the concepts you should be familiar with as you start using other LabVIEW features. Refer to the **Fundamentals** book on the **Contents** tab in the *LabVIEW Help* for more information about these concepts. The **Concepts** books contain information about LabVIEW programming concepts, and the **How-To** books contain step-by-step instructions for using LabVIEW.

All Controls and Indicators

The controls and indicators located on the **Express** subpalette of the **Controls** palette are a subset of the complete set of built-in controls and indicators available in LabVIEW. You can find all the controls and indicators that you can use to create the front panel on other subpalettes. However, subpalettes other than the **Express** subpalette categorize controls and indicators by functionality instead of having a subpalette for controls and a subpalette for indicators.

For example, the top level of the **Express** subpalette has a **Numeric Controls** subpalette and a **Numeric Indicators** subpalette. On the **Modern** and **Classic** subpalettes, these controls and indicators are located on the **Numeric** subpalette because they are all numeric objects.

Click the **View** button on the pinned **Controls** palette and select **Always Visible Categories»Show All Categories** from the shortcut menu to display all categories on the **Controls** palette.

Refer to the **Fundamentals»Building the Front Panel** book on the **Contents** tab in the *LabVIEW Help* for more information about using the complete set of built-in controls and indicators available in LabVIEW.

All VIs and Functions

The Express VIs and structures located on the **Express** subpalette of the **Functions** palette are a subset of the complete set of built-in VIs, functions, and structures available in LabVIEW.

Click the **View** button on the pinned **Functions** palette and select **Always Visible Categories»Show All Categories** from the shortcut menu to display all categories on the **Functions** palette.

LabVIEW uses colored icons to distinguish between functions, VIs, and Express VIs. Icons for functions have pale yellow backgrounds, most icons for VIs have white backgrounds, and icons for Express VIs appear surrounded by pale blue fields.

Express VIs appear on the block diagram as expandable nodes with icons surrounded by a blue field. Unlike Express VIs, most functions and VIs on the block diagram appear as icons rather than expandable nodes.

VIs

When you place a VI on the block diagram, the VI is a subVI. When you double-click a subVI, its front panel appears, rather than a dialog box in which you can configure options.

The icon for a VI appears in the upper right corner of the front panel and block diagram. This icon is the same as the icon that appears when you place the VI on the block diagram.

You can use a VI you create as a subVI. Refer to the **Fundamentals»Creating VIs and SubVIs** book on the **Contents** tab in the *LabVIEW Help* for more information about creating VIs and configuring them as subVIs.

You also can save the configuration of an Express VI as a subVI. Refer to the **Fundamentals»Building the Block Diagram** book on the **Contents** tab in the *LabVIEW Help* for more information about creating subVIs from Express VIs.

Functions

Functions are the fundamental operating elements of LabVIEW. Unlike VIs, functions do not have front panels or block diagrams.

Data Types

On the block diagram of a VI, the terminals for the front panel objects are different colors. The color and symbol of a terminal indicate the data type of the corresponding control or indicator. Colors also indicate the data types of wires, inputs, and outputs. The color of inputs and outputs of Express VIs indicate what type of data the input or output accepts or returns.

Data types indicate which objects, inputs, and outputs you can wire together. For example, a switch has a green border, so you can wire a switch to any Express VI input with a green label. A knob has an orange border, so you can wire a knob to any Express VI input with an orange label. However, you cannot wire a knob to an input with a green label. The wires you create are the same color as the terminal.

Refer to the **Fundamentals»Building the Block Diagram** book on the **Contents** tab in the *LabVIEW Help* for more information about data types.

Dynamic Data Type



Dynamic data stores the information generated or acquired by an Express VI. The dynamic data type appears as a dark blue terminal, shown at left. Most Express VIs accept or return dynamic data. You can wire dynamic data to any indicator or input that accepts numeric, waveform, or Boolean data. Wire dynamic data to an indicator that can best present the data. Such indicators include graphs, charts, and numeric indicators.

Most other VIs and functions in LabVIEW do not accept dynamic data. To use a built-in VI or function to analyze or process dynamic data, you must convert the dynamic data to numeric, Boolean, waveform, or array data.

Use the Convert from Dynamic Data Express VI to convert dynamic data to numeric, Boolean, waveform, and array data for use with other VIs and functions. When you wire dynamic data to an array indicator, LabVIEW inserts the Convert from Dynamic Data Express VI on the block diagram.

Use the Convert to Dynamic Data Express VI to convert numeric, Boolean, waveform, and array data to dynamic data for use with Express VIs.

Refer to the **Fundamentals»Building the Block Diagram** book on the **Contents** tab in the *LabVIEW Help* for more information about dynamic data types.

When to Use Other LabVIEW Features

The Express VIs, structures, and controls and indicators located on the **Express** subpalettes of the **Controls** and **Functions** palettes provide the functionality you need to build common measurement applications. The following list describes the applications that require you to use the VIs, functions, structures, controls, and indicators located on subpalettes other than the **Express** subpalette.

- **Programmatically control properties and methods for the LabVIEW environment, VIs, and controls and indicators**—You can control programmatically how a VI behaves when it runs, set the appearance of a control or indicator, or control how the LabVIEW environment behaves. Refer to the **Fundamentals»Programmatically Controlling VIs** book on the **Contents** tab in the *LabVIEW Help* for more information about these features.
- **Call code written in text-based languages**—You can use LabVIEW to communicate with applications written in a text-based programming language, such as C or C++. Refer to the **Fundamentals»Calling Code Written in Text-Based Programming Languages** book on the **Contents** tab in the *LabVIEW Help* for more information about these features.
- **Communicate with VIs across a network**—You can call a VI that resides on another computer running LabVIEW. Refer to the **Fundamentals»Networking in LabVIEW** book on the **Contents** tab in the *LabVIEW Help* for more information about these features.
- **Publish VIs on the Web**—You can publish the front panel of any VI on the Web, where users can interact with the front panel. Refer to the **Fundamentals»Networking in LabVIEW** book on the **Contents** tab in the *LabVIEW Help* for more information about these features.
- **Save data to a variety of file formats**—In addition to the text-based measurement file format, you can create files that other applications can use, such as text files and spreadsheet files. Refer to the **Fundamentals»File I/O** book on the **Contents** tab in the *LabVIEW Help* for more information about these features.
- **Customize menus**—You can configure which menu items appear when a user runs a VI. You also can create custom menus. Refer to the **Fundamentals»Creating VIs and SubVIs** book on the **Contents** tab in the *LabVIEW Help* for more information about these features.
- **Use LabVIEW projects**—You can use projects to group together LabVIEW files and non-LabVIEW files, create build specifications, and deploy or download files to multiple targets from one location. You

must use a project to build applications and shared libraries. You also must use a project to work with an RT, FPGA, or PDA target. Refer to the specific module documentation for more information about using projects with the RT, FPGA, and PDA Modules. Refer to the **Fundamentals»Organizing and Managing a Project** book on the **Contents** tab in the *LabVIEW Help* for more information about using LabVIEW projects.



Technical Support and Professional Services

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

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

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

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

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. 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.

Glossary

A

automatic scaling Ability of scales to adjust to the range of plotted values. On graph scales, autoscaling determines maximum and minimum scale values.

B

block diagram Pictorial description or representation of a program or algorithm. The block diagram consists of executable icons called nodes and wires that carry data between the nodes. The block diagram is the source code for the VI. The block diagram resides in the block diagram window of the VI.

Boolean controls and indicators Front panel objects to manipulate and display Boolean (TRUE or FALSE) data.

broken **Run** button Button that replaces the **Run** button when a VI cannot run because of errors.

broken VI VI that cannot run because of errors; signified by a broken arrow in the broken **Run** button.

C

channel	<p>1. Physical—a terminal or pin at which you can measure or generate an analog or digital signal. A single physical channel can include more than one terminal, as in the case of a differential analog input channel or a digital port of eight lines. The name used for a counter physical channel is an exception because that physical channel name is not the name of the terminal where the counter measures or generates the digital signal.</p> <p>2. Virtual—a collection of property settings that can include a name, a physical channel, input terminal connections, the type of measurement or generation, and scaling information. You can define NI-DAQmx virtual channels outside a task (global) or inside a task (local). Configuring virtual channels is optional in Traditional NI-DAQ and earlier versions but is integral to every measurement you take in NI-DAQmx. In Traditional NI-DAQ, you configure virtual channels in MAX. In NI-DAQmx, you can configure virtual channels either in MAX or in a program, and you can configure channels as part of a task or separately.</p> <p>3. Switch—a switch channel represents any connection point on a switch. It can be made up of one or more signal wires (commonly one, two, or four), depending on the switch topology. A virtual channel cannot be created with a switch channel. Switch channels may be used only in the NI-DAQmx Switch functions and VIs.</p>
checkbox	Small square box in a dialog box which you can select or clear. Checkboxes generally are associated with multiple options that you can set. You can select more than one checkbox.
conditional terminal	Terminal of a While Loop that contains a Boolean value that determines if the VI performs another iteration.
Context Help window	Window that displays basic information about LabVIEW objects when you move the cursor over each object. Objects with context help information include VIs, functions, constants, structures, palettes, properties, methods, events, and dialog box components.
control	Front panel object for entering data to a VI interactively or to a subVI programmatically, such as a knob, push button, or dial.
Controls palette	Palette that contains front panel controls, indicators, and decorative objects.
current VI	VI whose front panel, block diagram, or Icon Editor is the active window.

D

DAQ	<i>See</i> data acquisition (DAQ).
DAQ Assistant	A graphical interface for configuring measurement tasks, channels, and scales.
DAQ device	A device that acquires or generates data and can contain multiple channels and conversion devices. DAQ devices include plug-in devices, PCMCIA cards, and DAQPad devices, which connect to a computer USB or 1394 (FireWire) port. SCXI modules are considered DAQ devices.
data acquisition (DAQ)	<ol style="list-style-type: none">1. Acquiring and measuring analog or digital electrical signals from sensors, acquisition transducers, and test probes or fixtures.2. Generating analog or digital electrical signals.
data flow	Programming system that consists of executable nodes that execute only when they receive all required input data. The nodes produce output data automatically when they execute. LabVIEW is a dataflow system. The movement of data through the nodes determines the execution order of the VIs and functions on the block diagram.
data type	Format for information. In LabVIEW, acceptable data types for most VIs and functions are numeric, array, string, Boolean, path, refnum, enumeration, waveform, and cluster.
default	Preset value. Many VI inputs use a default value if you do not specify a value.
device	An instrument or controller you can access as a single entity that controls or monitors real-world I/O points. A device often is connected to a host computer through some type of communication network. <i>See also</i> DAQ device and measurement device .
drag	To use the cursor on the screen to select, move, copy, or delete objects.
driver	Software that controls a specific hardware device, such as a DAQ device.

E

- error message Indication of a software or hardware malfunction or of an unacceptable data entry attempt.
- Express VI A subVI designed to aid in common measurement tasks. You configure an Express VI using a configuration dialog box.

F

- For Loop Iterative loop structure that executes its subdiagram a set number of times. Equivalent to text-based code: `For i = 0 to n - 1, do...`
- front panel Interactive user interface of a VI. Front panel appearance imitates physical instruments, such as oscilloscopes and multimeters.
- function Built-in execution element, comparable to an operator, function, or statement in a text-based programming language.
- Functions** palette Palette that contains VIs, functions, block diagram structures, and constants.

G

- General Purpose Interface Bus GPIB. Synonymous with HP-IB. The standard bus used for controlling electronic instruments with a computer. Also called IEEE 488 bus because it is defined by ANSI/IEEE Standards 488-1978, 488.1-1987, and 488.2-1992.
- graph 2D display of one or more plots. A graph receives and plots data as a block.

I

- I/O Input/Output. The transfer of data to or from a computer system involving communications channels, operator input devices, and/or data acquisition and control interfaces.
- icon Graphical representation of a node on a block diagram.
- indicator Front panel object that displays output, such as a graph or LED.

instrument driver A set of high-level functions that control and communicate with instrument hardware in a system.

L

label Text object used to name or describe objects or regions on the front panel or block diagram.

LabVIEW Laboratory Virtual Instrument Engineering Workbench. LabVIEW is a graphical programming language that uses icons instead of lines of text to create programs.

LED Light-emitting diode.

legend Object a graph or chart owns to display the names and plot styles of plots on that graph or chart.

M

MAX *See* Measurement & Automation Explorer.

Measurement & Automation Explorer The standard National Instruments hardware configuration and diagnostic environment for Windows.

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

menu bar Horizontal bar that lists the names of the main menus of an application. The menu bar appears below the title bar of a window. Each application has a menu bar that is distinct for that application, although some menus and commands are common to many applications.

N

NI-DAQ Driver software included with all NI DAQ devices and signal conditioning components. NI-DAQ is an extensive library of VIs and ANSI C functions you can call from an application development environment (ADE), such as LabVIEW, to program an NI measurement device, such as the M Series multifunction I/O (MIO) DAQ devices, signal conditioning modules, and switch modules.

NI-DAQmx	The latest NI-DAQ driver with new VIs, functions, and development tools for controlling measurement devices. The advantages of NI-DAQmx over earlier versions of NI-DAQ include the DAQ Assistant for configuring channels and measurement tasks for your device for use in LabVIEW, LabWindows™/CVI™, and Measurement Studio; NI-DAQmx simulation for most supported devices for testing and modifying applications without plugging in hardware; and a simpler, more intuitive API for creating DAQ applications using fewer functions and VIs than earlier versions of NI-DAQ.
node	Program execution element. Nodes are analogous to statements, operators, functions, and subroutines in text-based programming languages. On a block diagram, nodes include functions, structures, and subVIs.
numeric controls and indicators	Front panel objects to manipulate and display numeric data.
O	
object	Generic term for any item on the front panel or block diagram, including controls, indicators, structures, nodes, wires, and imported pictures.
Operating tool	Tool to enter data into controls or to operate them.
P	
palette	Displays objects or tools you can use to build the front panel or block diagram.
plot	Graphical representation of an array of data shown either on a graph or a chart.
Positioning tool	Tool to move and resize objects.
pull-down menus	Menus accessed from a menu bar. Pull-down menu items are usually general in nature.
PXI	PCI eXtensions for Instrumentation. A modular, computer-based instrumentation platform.

R

RMS Root Mean Square.

S

sample Single analog or digital input or output data point.

scale Part of graph, chart, and some numeric controls and indicators that contains a series of marks or points at known intervals to denote units of measure.

shortcut menu Menu accessed by right-clicking an object. Menu items pertain to that object specifically.

string Representation of a value as text.

structure Program control element, such as a Flat Sequence structure, Stacked Sequence structure, Case structure, For Loop, or While Loop.

subVI VI used on the block diagram of another VI. Comparable to a subroutine.

T

task A collection of one or more channels, timing, triggering, and other properties in NI-DAQmx. A task represents a measurement or generation you want to perform.

terminal Object or region on a node through which data pass.

tip strip Small yellow text banners that identify the terminal name and make it easier to identify terminals for wiring.

tool Special cursor to perform specific operations.

toolbar Bar that contains command buttons to run and debug VIs.

Traditional NI-DAQ
(Legacy) An older driver with outdated APIs for developing data acquisition, instrumentation, and control applications for older National Instruments DAQ devices. You should use Traditional NI-DAQ (Legacy) only in certain circumstances. Refer to the *NI-DAQ Readme* for more information about when to use Traditional NI-DAQ (Legacy), including a complete list of supported devices, operating systems, and application software and language versions.

V

VI	<i>See</i> virtual instrument (VI).
virtual instrument (VI)	Program in LabVIEW that models the appearance and function of a physical instrument.
VXI	VME eXtensions for Instrumentation (bus).

W

waveform	Multiple voltage readings taken at a specific sampling rate.
waveform chart	Indicator that plots data points at a certain rate.
While Loop	Loop structure that repeats a section of code until a condition occurs.
wire	Data path between nodes.
Wiring tool	Tool to define data paths between terminals.

Index

Symbols

- .lvn files, 2-9, 2-10, 2-13
- .tdm files, 2-9, 2-10, 2-13

A

- acquiring
 - information for instruments, 4-7
 - signals, 4-1
- Acquiring a Signal VI block diagram (figure), 1-12
- adding
 - channels to tasks, 4-5
 - controls from the block diagram, 3-4
 - controls to the front panel, 1-5
 - graph indicators, 3-4
 - inputs to Express VIs, 1-8, 3-4
 - numeric indicators, 3-4
 - visual cues on front panel, 2-6
 - warning lights, 2-6
- Amplitude and Level Measurements Express VI, 2-2
 - analyzing voltage, 2-5
- analyzing signals, 2-5
- applications, building, 5-4
- Arithmetic & Comparison palette, 1-10

B

- block diagram, 1-5, 1-17
 - indicators, 2-13
 - modifying, 2-3
 - showing, 1-6
- broken
 - Run button, 2-4
 - wires, 2-4, 2-14
- Build Table Express VI, 3-8

- building
 - applications, 5-4
 - shared libraries, 5-4
- button
 - adding, 2-10
 - placing on block diagram, 3-10
 - Run, 1-5

C

- calling code from text-based languages, 5-4
- changing signal types, 1-6
- channels, 4-2
 - adding to tasks, 4-5
 - renaming, 4-4
- communicating
 - with instruments, 4-6, 4-10
 - with LabVIEW applications across networks, 5-4
- Comparison Express VI, setting warning level, 2-7
- configuration dialog boxes, 1-17
- Context Help window, 2-2, 2-13, 3-2, 3-9
 - button, 3-2
 - displaying configuration of Express VIs, 2-2
 - displaying errors, 2-14
 - figure, 2-2
- controlling execution speed, 3-6, 3-7, 3-11
- controls, 1-16, 2-13, 5-1
 - adding from the block diagram, 3-4
 - adding to the front panel, 1-5
 - configuring, 1-17
 - creating, 3-4, 3-10
 - customizing, 1-13
 - data types, 5-3
 - numeric, 1-6, 5-1
 - palette, 1-5

- Controls palette, 1-5
 - figure, 1-6
 - showing all categories, 5-1
- conventions used in this manual, *ix*
- Convert from/to Dynamic Data Express VIs, 5-3
- creating
 - controls, 3-4, 3-10
 - graph indicators, 3-4
 - indicators, 3-10
 - NI-DAQmx tasks, 4-2
- customizing
 - block diagrams, 3-10
 - controls, 1-13
 - front panels, 3-4
 - indicators, 1-15
 - menus, 5-4
 - simulated signals, 2-3

D

- DAQ Assistant Express VI, 4-2, 4-8
- DAQ devices, 4-2
- data
 - saving
 - to a file, 2-10
 - when prompted by a user, 2-11
 - storing, 2-10
- data flow, 1-9, 1-12, 1-17
- data types, dynamic, 5-3
- deleting wires, 1-10
- deselecting objects, 1-8
- diagnostic tools (NI resources), A-1
- displaying
 - data from DAQ devices, 4-4
 - data in tables, 3-8, 3-11
 - signals in a graph, 1-12

- documentation
 - conventions used in this manual, *ix*
 - introduction to this manual, *ix*
 - NI resources, A-1
 - related documentation, *x*
- drivers
 - instrument, 4-6, 4-9
 - NI resources, A-1
- dynamic data, 5-3
 - converting from and to, 5-3

E

- Error list window, 2-4, 2-14
- errors, 2-14
 - displaying, 2-4
 - displaying in Context Help window, 2-14
 - list, 2-4, 2-14
 - window, 2-4, 2-14
- example VIs
 - NI Example Finder, 3-7
 - NI resources, A-1
- Execution Control palette, 3-5, 3-11
- execution speed, controlling, 3-6, 3-7, 3-11
- Express VIs, 1-17
 - Amplitude and Level Measurements, 2-2, 2-5
 - Build Table, 3-8
 - Comparison, 2-7
 - configuration dialog boxes, 1-17
 - Convert from/to Dynamic Data, 5-3
 - DAQ Assistant, 4-2, 4-8
 - dynamic data, 5-3
 - inputs, 1-17
 - Instrument I/O Assistant, 4-6, 4-9
 - outputs, 1-17
 - Scaling and Mapping, 1-10
 - Simulate Signals, 1-6
 - Time Delay, 3-7
 - Write To Measurement File, 2-9, 2-13

F

- files
 - grouping, 5-4
 - saving to other formats, 5-4
- front panel, 1-5, 1-16
 - Acquiring a Signal VI (figure), 1-2
 - adding
 - controls, 1-5
 - visual cues, 2-6
 - controls, 1-16, 2-13
 - customizing, 3-4
 - indicators, 1-17
 - modifying, 2-4
 - showing, 1-9
 - Warning Light VI (figure), 2-1
- functions, 5-2
 - Merge Signals, 1-12, 3-4
- Functions palette
 - figure, 1-10
 - showing all categories, 5-2

G

- Getting Started window, 1-3, 2-2
 - figure, 1-3
- graph indicators, creating, 3-4
- graphing
 - data from DAQ devices, 4-4
 - two signals, 1-12
- grouping files, 5-4

H

- help
 - button, 2-5
 - Context Help window, 2-2, 2-13, 3-2, 3-9
 - LabVIEW Help*, 2-5, 2-13, 3-9
 - LabVIEW resources, 2-13, 3-9
 - searching, 3-3, 3-6, 3-10
 - technical support, A-1

I

- indicators, 1-17, 2-13, 5-1
 - adding numeric, 3-4
 - configuring, 1-17
 - creating, 3-10
 - customizing, 1-15
 - data type, 5-3
 - numeric, 5-1
 - removing, 2-4
- Input palette, 3-2
- Instrument Driver Network, 4-6, 4-9
- instrument drivers, 4-6, 4-9
 - NI resources, A-1
- Instrument I/O Assistant Express VI, 4-6, 4-9
- instruments
 - acquiring information, 4-7
 - communicating, 4-6, 4-10
 - parsing information, 4-7
 - selecting, 4-7

K

- Knob Control, customizing (figure), 1-14
- KnowledgeBase, A-1

L

- LabVIEW
 - help resources, using, 3-9
 - other features, using, 5-1
 - projects, 5-4
- LabVIEW Help*, 2-13, 3-9
 - searching for Express VIs, 3-3
- LEDs, palette, 2-6
 - figure, 2-6
- LVM. *See* .lvm files

M

manual. *See* documentation
 marquee, 3-8
 menus, customizing, 5-4
 Merge Signals function, 1-12, 3-4
 figure, 1-13
 modifying
 block diagrams, 2-3
 front panels, 2-4
 signals, 1-10, 3-3

N

National Instruments support and services, A-1
 New dialog box, 1-3, 1-16, 2-1, 2-2
 figure, 1-4
 NI Example Finder, 3-7
 NI Instrument Driver Network, 4-6, 4-9
 NI support and services, A-1
 NI-DAQmx
 creating tasks, 4-2
 tasks, 4-2, 4-9
 testing tasks, 4-4
 Numeric Controls palette, 1-6

O

objects, deselecting, 1-8
 Operating tool, 1-9

P

palettes
 Arithmetic & Comparison, 1-10
 Controls, 1-5
 Execution Control, 3-5, 3-11
 Functions, 1-10
 Input, 3-2
 Numeric Controls, 1-6
 searching, 2-7, 2-10

 showing all categories, 5-1, 5-2
 Text Indicators, 3-8
 parsing information for instruments, 4-7
 placing objects on the block diagram from the help, 3-10
 Positioning tool, 1-8
 programatically controlling VIs, 5-4
 programming examples (NI resources), A-1
 projects, 5-4
 property dialog boxes, 1-17
 publishing VIs on the Web, 5-4

R

related documentation, *x*
 Run button, 1-5, 1-9
 broken, 2-4, 2-14
 running VIs, 1-9
 continuously, 3-5

S

saving data
 different file formats, 5-4
 Save Data VI block diagram (figure), 2-12
 to files, 2-9, 2-10, 2-13
 when prompted by user, 2-11, 2-12
 Scaling and Mapping Express VI, 1-10
 defining slope, 1-11
 searching
 examples, 3-7
 help, 3-3, 3-6, 3-10
 palettes, 2-7, 2-10
 selecting
 instruments, 4-7
 objects, 1-8
 shared libraries, building, 5-4

signals

- acquiring, 4-1
- analyzing, 2-5
- changing type, 1-6
- modifying, 1-10, 3-3

Simulate Signal Express VI, 1-6

software (NI resources), A-1

T

tables, 3-8

- displaying data, 3-11

tasks

- adding new channels, 4-5
- NI-DAQmx, 4-9
- testing, 4-4

TDM. *See* .tdm files

technical support, A-1

template VIs, 1-2, 1-3, 1-16

Text Indicators palette, 3-8

text-based languages, calling code, 5-4

Time Delay Express VI, 3-7

tools

- Operating, 1-9
- Positioning, 1-8
- Wiring, 1-8

training and certification (NI resources), A-1

troubleshooting (NI resources), A-1

U

user interface. *See* front panel

V

virtual instruments. *See* VIs

VIs, 1-1, 5-2

- blank, 3-1
- building, 1-1, 2-1
- customizing menus, 5-4
- icons, 5-2
- new, 3-1
- programmatically controlling, 5-4
- publishing on the Web, 5-4
- running, 1-9
- subVIs, 5-2
- template, 1-2, 1-3, 1-16

W

Warning Light VI block diagram (figure), 2-8

Web resources, A-1

While Loop, 3-6

wires

- broken, 2-4, 2-14
- deleting, 1-10

wiring

- objects on the block diagram, 1-8
- tool, 1-8

Write to Measurement File Express VI, 2-9,
2-10, 2-13

- saving data, 2-10