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 IS 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 INDIRECT 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 maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, acts 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, by any means, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

For copyright notices, conditions, and disclaimers, including information regarding certain third-party components used in LabVIEW, refer to the Copyright topic of the LabVIEW Help.

Trademarks
CVI, LabVIEW, National Instruments, NI, ni.com, the National Instruments corporate logo, and the Eagle logo are trademarks of National Instruments Corporation. Refer to the Trademark Information at ni.com/ trademarks for other National Instruments trademarks.

The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States 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/technology, refer to the appropriate location: Help»Patents in your software, file on your media, or the National Instruments Patent Notice at 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 CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHERE 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 INTESTS, INTESTS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP APPLICATION, INSTALLATION ERRORS, SOFTWARE HARDWARE COMPATIBILITY PROBLEMS, SOFTWARE SOFTWARE INTESTS OR FAILURES OF ELECTRONIC MONITORS OR CONTROL DEVICES, TRANSMISSION FAILURES OF ELECTRONIC SYSTEMS HARDWARE AND/OR SOFTWARE, UNANTICIPATED USES OR MISUSES. OR EREDOCS ON THE PART OF THE USER OR APPLICATION DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREFORER 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 REASONABLE PRECAUTIONARY 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
## Contents

<table>
<thead>
<tr>
<th>Chapter 3</th>
<th>Analyzing and Saving a Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building an Analysis VI</td>
<td>3-2</td>
</tr>
<tr>
<td>Modifying a VI Created from a Template</td>
<td>3-2</td>
</tr>
<tr>
<td>Adding a Signal</td>
<td>3-4</td>
</tr>
<tr>
<td>Adding Two Signals</td>
<td>3-5</td>
</tr>
<tr>
<td>Filtering a Signal</td>
<td>3-7</td>
</tr>
<tr>
<td>Modifying the Appearance of Graphs</td>
<td>3-8</td>
</tr>
<tr>
<td>Analyzing the Amplitude of a Signal</td>
<td>3-9</td>
</tr>
<tr>
<td>Controlling the Speed of Execution</td>
<td>3-9</td>
</tr>
<tr>
<td>Setting a Warning Level Limit</td>
<td>3-11</td>
</tr>
<tr>
<td>Warning the User</td>
<td>3-12</td>
</tr>
<tr>
<td>Configuring a VI to Save Data to a File</td>
<td>3-13</td>
</tr>
<tr>
<td>Saving Data to a File</td>
<td>3-14</td>
</tr>
<tr>
<td>Adding a Button That Stores Data When Chicked</td>
<td>3-14</td>
</tr>
<tr>
<td>Saving Data When Prompted by a User</td>
<td>3-15</td>
</tr>
<tr>
<td>Viewing Saved Data</td>
<td>3-16</td>
</tr>
<tr>
<td>Controls and Indicators</td>
<td>3-17</td>
</tr>
<tr>
<td>Filtering Data</td>
<td>3-17</td>
</tr>
<tr>
<td>Saving Data</td>
<td>3-17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 4</th>
<th>Hardware: Acquiring Data and Communicating with Instruments (Windows)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware and Software Requirements</td>
<td>4-1</td>
</tr>
<tr>
<td>Acquiring a Signal in NI-DAQmx</td>
<td>4-2</td>
</tr>
<tr>
<td>Creating an NI-DAQmx Task</td>
<td>4-2</td>
</tr>
<tr>
<td>Graphing Data from a DAQ Device</td>
<td>4-5</td>
</tr>
</tbody>
</table>

---

## About This Manual

### Conventions

- **Using a Table to Display Data**: 2-9
- **Searching for Examples**: 2-11
- **Summary**: 2-12
- **Using the LabVIEW Help Resources**: 2-12
- **Customizing the Block Diagram Code**: 2-13
- **Creating Controls and Indicators**: 2-13
- **Controlling When a VI Stops Running**: 2-14
- **Errors and Broken Wires**: 2-14
- **Displaying Data in a Table**: 2-14
- **Using the NI Example Finder**: 2-14
- **Shortcuts**: 2-15

### Related Documentation

- **Using the LabVIEW Help Resources**: ix
- **Related Documentation**: x

### Chapter 1

**Getting Started with LabVIEW Virtual Instruments**

- Building a Virtual Instrument: 1-2
- Launching LabVIEW: 1-3
- Opening a New VI from a Template: 1-4
- Adding a Control to the Front Panel: 1-6
- Changing a Signal Type: 1-8
- Wiring Objects on the Block Diagram: 1-10
- Running a VI: 1-11
- Modifying a Signal: 1-11
- Displaying Two Signals on a Graph: 1-15
- Customizing a Knob Control: 1-16
- Customizing a Waveform Graph: 1-17
- Summary: 1-18
- New Dialog Box and Template VIs: 1-18
- Front Panel: 1-19
- Block Diagram: 1-19
- Front Panel and Block Diagram Tools: 1-19
- Running and Stopping a VI: 1-20
- Express VIs: 1-20
- LabVIEW Documentation Resources: 1-20
- Property Dialog Boxes: 1-21
- Shortcuts: 1-21

### Chapter 2

**Customizing a VI**

- Building a VI from a Blank VI: 2-1
- Opening a Blank VI: 2-2
- Adding an Express VI That Simulates a Signal: 2-2
- Searching the Help and Modifying a Signal: 2-3
- Customizing a User Interface from the Block Diagram: 2-5
- Configuring a VI to Run Continuously until the User Stops It: 2-6
- Using the Error List Window: 2-8
- Controlling the Speed of Execution: 2-9
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 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, dialog boxes, sections of dialog boxes, menu names, and palette names.

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

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.
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 front panel, 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.

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 the LabVIEW Help to access information about LabVIEW programming concepts, step-by-step instructions for using LabVIEW, and reference information about LabVIEW VIs, functions, palettes, menus, tools, properties, methods, events, dialog boxes, and so on. The LabVIEW Help also lists the LabVIEW documentation resources available from National Instruments. Access the LabVIEW Help by selecting Help»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.
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](image)

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

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

Building a Virtual Instrument
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 in the front panel window.

1. Launch LabVIEW.
2. In the Getting Started window, click the New or VI from Template link to display the New dialog box.
3. 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.

4. 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 a template.

LabVIEW displays two windows: the front panel window and the block diagram window.

5. Examine the front panel window.

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.
2. If you are a new LabVIEW user, the Controls palette opens with the Express palette, shown in Figure 1-4, visible by default. If you do not see the Express palette, click Express on the Controls palette to display the Express palette.

![Controls Palette](image)

Figure 1-4. Controls Palette

3. Move the cursor over the icons on the Express 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.

Note Some palette objects display a short name on the palette that is different from the name that appears in the tip strip. The short name abbreviates the name of the palette object so that it fits in the space available on the palette. If you have difficulty finding a palette object by its short name, use the Search button on the Controls or Functions palette to find the palette object by name.

4. Click the Numeric Controls icon to display the Numeric Controls palette.

5. Click the Knob control on the Numeric Controls palette to attach the control to the cursor, and then add the knob to 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.

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

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.

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

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

Adding a Control to the Front Panel

Front panel controls 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 in the front panel window, select View»Controls Palette.

Tip You can right-click any blank space in the front panel or the block diagram to display a temporary version of the Controls or Functions palette. The Controls or Functions palette appears with a thumbtack icon in the upper left corner. Click the thumbtack to pin the palette so it is no longer temporary.
Changing a Signal Type

The block diagram has a blue icon labeled Simulate Signal. This icon represents the Simulate Signal Express VI. 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 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.

2. Right-click the Simulate Signal Express VI and select Properties from the shortcut menu to display the Configure Simulate Signal dialog box.

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

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 inputs and outputs 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.

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.

Tip: You also can double-click the Express VI to display the Configure Simulate Signal dialog box.
Running a VI

Running a VI executes the solution.

Complete the following steps to run the Acquiring a Signal VI.
1. Display the front panel by pressing the <Ctrl-E> keys or by clicking the front panel.
2. Click the Run button or press the <Ctrl-R> keys to run the VI.
3. Move the cursor over the knob.
The cursor becomes a hand, or the Operating tool, shown at left. Use the Operating tool to change the value of a control.
4. 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. As you change the amplitude, the Operating tool displays a tip strip that indicates the numeric value of the knob. The y-axis on the graph scales automatically to account for the change in amplitude.
5. Click the STOP button, shown at left, to stop the VI.
The STOP button stops the VI after the loop completes its 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 the VIs you create with a stop button to avoid this problem.

Modifying a Signal

Complete the following steps to scale the signal by 10 and display the results in the front panel graph.
1. In the block diagram, use the Positioning tool to 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.

Wiring Objects on the Block Diagram

To use the knob 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.

Note: You can resize only loops and structures on the block diagram. Go to the front panel to resize objects you have added to the front panel.
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 loop, 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.
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.
5. 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.
6. Select File→Save to save the VI.
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 palette visible by default. If you have selected another palette, you can return to the Express palette by clicking Express on the Functions palette.

4. On the Arithmetic & Comparison palette, select the Formula Express VI, shown at left, and place it on the block diagram 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. The Configure Formula dialog box appears when you place the Express VI on the block diagram. When you place an Express VI on the block diagram, the configuration dialog box for that VI always appears automatically.

   Note If you place an object too close to another object on the block diagram, auto wiring might wire the two objects together. Delete the wires if the automatic wiring is wrong. To configure auto wiring select Tools»Options then select Block Diagram from the Category list. Remove the checkmark from the Enable auto wiring checkbox to turn off auto wiring.

5. Click the Help button, shown at left, in the bottom right corner of the Configure Formula dialog box to display the LabVIEW Help topic for this Express VI. The Formula help topic describes the Express VI, the configuration dialog box options, and the inputs and outputs of the Express VI. Each Express VI has a corresponding help topic you can access by clicking the Help button in the configuration dialog box or by right-clicking the Express VI and selecting Help from the shortcut menu.
Displaying Two Signals on a Graph

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

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

1. In the block diagram, move the cursor over the arrow on the **Sawtooth** output of the Simulate Signal Express VI.
2. 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](image)

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 original sawtooth wave and the scaled sawtooth wave with 10 times the amplitude, as you specified in the Formula Express VI. The maximum value on the y-axis automatically scales as you turn the knob.
5. Click the **STOP** button to stop the VI.

**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. You can also click the **Clean Up Diagram** button on the block diagram toolbar to have LabVIEW automatically reroute all existing wires and rearrange objects on the block diagram to generate a cleaner look.

14. Press the <Ctrl-S> keys or select **File»Save** to save the VI.
Chapter 1  Getting Started with LabVIEW Virtual Instruments

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.

1. Right-click the front panel knob and select **Properties** from the shortcut menu to display the **Knob Properties** dialog box. Click the **Appearance** tab to display the **Appearance** page.
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.
3. Click the **Scale** tab. In the **Scale Style** section, place a checkmark in the **Show color ramp** checkbox.
4. The knob in the front panel window updates to reflect these changes.
5. Click the **OK** button to save the current configuration and close the **Knob Properties** dialog box.
6. Save the VI.

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. In the front panel window, 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 to add one item to the legend. When you release the mouse button, the second plot name appears.
3. Right-click the waveform graph and select **Properties** from the shortcut menu to display the **Graph Properties** dialog box.

Figure 1-11. Expanding a Plot Legend

Figure 1-10. Knob Properties Dialog Box

3. Click the **Scale** tab. In the **Scale Style** section, place a checkmark in the **Show color ramp** checkbox.

The knob in the front panel window 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.
**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.

**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 in the following ways: from controls to VIs and functions, from VIs and functions to indicators, and from VIs and functions to other VIs and functions. 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.

**Front Panel and Block Diagram Tools**

The Positioning tool appears when you move the cursor over an object in the front panel window or on the block diagram. The cursor becomes an arrow that you can use to select, position, and resize objects. The Wiring tool appears when you move the cursor over a terminal of a block diagram object. The cursor becomes a spool that you can use to connect objects on the block diagram through which you want data to flow.

4. On the Plots page, select Sawtooth from the top 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 (Formula Result) from the top pull-down menu.

6. Place a checkmark in the Do not use waveform names for plot names checkbox. This action lets you edit the labels on the graph.

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 Graph Properties dialog box. The plot color and plot legend change.

9. Reopen the 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 launch LabVIEW.
- Select Files-New from the menu bar of the Getting Started window, the front panel, or the block diagram.
Running and Stopping a VI

Running a VI executes the solution of the VI. Click the Run button or press the <Ctrl-R> keys to run a VI. The Run button changes to a darkened arrow to indicate the VI is running. You can stop a VI immediately by clicking the Abort Execution button. However, aborting a VI that uses external resources might leave the resources in an unknown state. Design the VIs you create with a stop button to avoid this problem. A stop button stops a VI after the VI completes its current iteration.

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.

LabVIEW Documentation Resources

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, tools, properties, methods, events, dialog boxes, and so on. The LabVIEW Help also lists the LabVIEW documentation resources available from National Instruments. To access help information for Express VIs, click the Help button in the configuration dialog box while you configure an Express VI. 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>LabVIEW Help.
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 structures and Express VIs 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 front panel table. After 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 45 minutes.
2. On the Functions palette, select the Express » Input palette and move the cursor over one of the Express VIs on the Input palette. When you move the cursor over a VI, the Context Help window displays information about that VI.
3. Use the information that appears in the Context Help window to find the Express VI that can simulate a sine wave signal. Keep the Context Help window open. The context help provides useful information as you complete the rest of this exercise.
4. Select the Express VI that can simulate a sine wave signal 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) and Amplitude. 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 displays information about how you configured the Simulate Signal Express VI in addition to the standard context help description.
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. On the block diagram, move the cursor over the Simulate Signal Express VI to display the Context Help window and click the Detailed Help link 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 » LabVIEW Help.
Chapter 2 Customizing a VI

Getting Started with LabVIEW

© National Instruments Corporation 2-5

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.

1. 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. A Mean indicator, shown at left, appears on the block diagram.

2. 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.

In previous exercises, you learned to add inputs and outputs by expanding the Express VI using the down arrows. Using the shortcut menu is a different way to display and select the inputs and outputs of an Express VI.

3. Right-click the Enable input and select Create»Control from the shortcut menu to create a switch. A Boolean control, shown at left, appears on the block diagram.

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.

4. 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.

5. 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.

6. Arrange the objects on the block diagram so they appear similar to Figure 2-3.

![Figure 2-3. Block Diagram of the Reduce Samples VI](image)

7. Display the front panel.

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

Note: You might need to scroll or resize the front panel to see all 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, and then stops running. 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. Display the block diagram.

3. Click the Search button, shown at left, on the Functions palette, and enter while in the text box. LabVIEW searches as you type the first few letters and displays any matches in the search results text box.

If there are objects with the same name, use the information in the brackets to the right of each object name to decide which object to select. Some objects are located on multiple palettes because you can use them for multiple applications.
Chapter 2 Customizing a VI

Getting Started with LabVIEW 2-8 ni.com © National Instruments Corporation

4. Double-click While Loop <<Execution Control>> to display the Execution Control subpalette and temporarily highlight the While Loop on the subpalette.
5. Select the While Loop on the Execution Control palette.
6. Move the cursor to the upper left corner of the block diagram. Click and drag the cursor diagonally to enclose off the Express VIs and wires, as shown in Figure 2-4.

![Figure 2-4. Placing the While Loop around the Express VIs](image)

7. Release the mouse to place the While Loop around the Express VIs and wires.
   
   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.
8. 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.
9. Click the STOP button and save the VI.

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

Using the Error List Window

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

Complete the following steps to remove the Mean indicator from the front panel.

1. Display the front panel and move the cursor over the Mean indicator until the Positioning tool appears.
2. Click the Mean 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. The dashed black line 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.
5. In the errors and warnings list, select the Wire: has loose ends error and click the Help button to display more information about the error.

Tip You also can 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.

6. In the errors and warnings list, double-click the Wire: has loose ends error to highlight the broken wire.
7. 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 wire.
8. Select View»Error List to display the Error list window. No errors appear in the errors and warnings field.

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

9. Click the Close button to close the Error list window.
   The Run button no longer appears broken.
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, shown at left, on the Functions palette and place it inside the While Loop. You can use the Time Delay Express VI to control the execution rate of the VI.
2. Enter 0.25 in the Time delay (seconds) text box. This time delay specifies how fast the loop runs. With a 0.25 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 front panel table.

1. Display the front panel.
2. On the Controls palette search for the Express Table indicator and add it to the right of the waveform graph.
3. Display the block diagram. LabVIEW wired the Table terminal to the Build Table Express VI.
4. 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, 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.
5. Drag the objects into the While Loop to the right of the Sample Compression Express VI.
   If you drag objects near the border of the While Loop, the loop resizes to enclose the Build Table Express VI and the Table terminal after you add the object to the block diagram.
6. Use the Wiring tool to wire the Mean output of the Sample Compression Express VI to the Signals input of the Build Table Express VI.
7. The block diagram should appear similar to Figure 2-5.

8. Display the front panel and arrange the controls and indicators as shown in Figure 2-1.
9. Run the VI.
10. Click the Enable switch.
    If the Enable switch is on, 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.
11. Stop the VI.
12. Experiment with properties of the table by using the Table Properties dialog box. For example, try changing the number of columns to one.
13. Save and close 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. In the Help section of the Getting Started window, select LabVIEW Help to display the LabVIEW Help.
2. Click the Search tab. In the Type in the word(s) to search for text box enter time delay express VI and press the <Enter> key. (Mac OS X and Linux) Choose the Full Text option in the Search Options section of the Search the LabVIEW Help dialog box to narrow the search results.

3. (Windows) Click the Location column header to sort the search results by content type. Reference topics contain reference information about LabVIEW objects such as VIs, functions, palettes, menus, and tools. 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 Express VI search result to display the reference 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. Run the VI and move the vertical pointer slides. The amplitudes of Value A and B change as you move the vertical pointer slides.
7. Stop the VI.
8. Select Window»Show Block Diagram and read the block diagram comments.
9. Close the example VI and open the LabVIEW Help window.
10. Click the Find related examples button to open the NI Example Finder and display a list of examples similar to the example that uses this VI. The NI Example Finder searches among hundreds of examples, including all installed examples and the examples located on the NI Developer Zone at ni.com/zone. You can modify an example to fit an application, or you can copy and paste from one or more examples into a VI that you create.

Note Always select File»Save As when you save a modified example to avoid overwriting the example program in 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. To launch the NI Example Finder and browse or search examples, select Help»Find Examples or click the Find Examples link in the Examples section of the Getting Started window.

Note There is not an example for all VIs.

11. After you experiment with the NI Example Finder and the example VIs, close the NI Example Finder.

Summary

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

Using the LabVIEW Help Resources

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, functions, structures, palettes, dialog box components, and so on. To access the Context Help window, select Help»Show Context Help or press the <Ctrl-H> keys. (Mac OS X) Press the <Command-Shift-H> keys.
- 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.
- The LabVIEW Help contains detailed information about LabVIEW objects. To access the LabVIEW Help topic for an object, move the cursor over the object and click the Detailed help link in the Context Help window. You also can right-click an object on the block diagram or on a pinned palette and select Help from the shortcut menu.
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. After 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.

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.

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

Using the NI Example Finder
Use the NI Example Finder to browse or search examples installed on your computer or on the NI Developer Zone at ni.com/zone. These examples demonstrate how to use LabVIEW to perform a wide variety of test, measurement, control, and design tasks. 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.

Customizing the Block Diagram Code
You also can use many controls, indicators, Express VIs, and structures to build a VI. You can customize VIs in many ways. The following examples review a few common ways to customize VIs, including creating controls and indicators, configuring when VI stops, correcting broken wires, and displaying 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.
Examples can show you how to use specific VIs or functions. 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. 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.

Always select File»Save As when you save a modified example to avoid accidentally overwriting the example program in the NI Example Finder.

### Shortcuts

This chapter introduced the following keyboard shortcuts.

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Ctrl-N&gt;</td>
<td>Opens a new, blank VI.</td>
</tr>
<tr>
<td>&lt;Ctrl-H&gt;</td>
<td>Shows or hides the Context Help window. <em>(Mac OS X)</em> Press the &lt;Command-Shift-H&gt; keys.</td>
</tr>
<tr>
<td>&lt;Ctrl-Space&gt;</td>
<td>Displays the Quick Drop dialog box. <em>(Mac OS X)</em> Press the &lt;Command-Shift-Space&gt; keys.</td>
</tr>
<tr>
<td>&lt;Ctrl-B&gt;</td>
<td>Deletes all broken wires in a VI.</td>
</tr>
<tr>
<td>&lt;Ctrl-L&gt;</td>
<td>Displays the Error list window.</td>
</tr>
</tbody>
</table>
Building an Analysis VI

In the following exercises, you will build a VI that generates a signal, filters the signal, indicates if the signal exceeds a certain limit, and records the data. After 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 40 minutes.

Modifying a VI Created from a Template

Complete the following steps to create a VI that generates, analyzes, and displays a signal.

1. In the Getting Started window, click New 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.
Chapter 3 Analyzing and Saving a Signal

Getting Started with LabVIEW

Adding a Signal

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 create an additional simulated signal that adds uniform white noise to the sine wave.

1. On the block diagram, use the Positioning tool to select the Simulate Signal Express VI. Hold down the <Ctrl> key and click and drag to create an additional Simulate Signal Express VI on the block diagram. (Mac OS X) Hold down the <Option> key and drag. (Linux) You also can hold down the middle mouse button and drag.

2. Release the mouse button to place the copied Simulate Signal Express VI below the original Simulate Signal Express VI. LabVIEW updates the name of the copied Simulate Signal Express VI to Simulate Signal2.

3. Double-click the Simulate Signal2 Express VI to display the Configure Simulate Signal dialog box.

4. Select Sine from the Signal type pull-down menu.

5. Enter 60 in the Frequency (Hz) text box.

6. Enter 0.1 in the Amplitude text box.

7. Place a checkmark in the Add noise checkbox to add noise to the sine signal.

8. Select Uniform White Noise from the Noise type pull-down menu.

9. Enter 0.1 in the Noise amplitude text box.

10. Enter -1 in the Seed number text box.

11. In the Timing section, select the Run as fast as possible option.

12. In the Signal Name section, remove the checkmark from the Use signal type name checkbox.

13. Enter 60 Hz and Noise in the Signal name text box.

When you change the signal name in the Configure Simulate Signal dialog box, LabVIEW changes the name of the signal output on the block diagram. Changing the signal name makes it easier for you to identify the signal type when you view the Express VI on the block diagram.

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

Figure 3-2. Configure Simulate Signal Dialog Box

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

Adding Two Signals

To add two signals together to create one signal, you can use the Formula Express VI. Rather than merging two signals on one graph, the Formula Express VI adds both signals together to create a single signal on the graph. You can use this Express VI to add noise to a signal.
Chapter 3 Analyzing and Saving a Signal

Filtering a Signal

You can use the Filter Express VI to process signals through filters and windows.

Complete the following steps to configure the Filter Express VI to filter the signal using an infinite impulse response (IIR) filter.

1. Display the block diagram window and remove the wire that connects the Result output of the Formula Express VI to the Signals input of the Amplitude and Level Measurements Express VI.
2. Remove all broken wires that result from removing the wire.
3. Search for the Filter Express VI shown at left, and add it to the block diagram between the Simulate Signal2 Express VI and the Amplitude and Level Measurements Express VI. The Configure Filter dialog box appears.
4. In the Filter Specifications section, change the Cutoff Frequency (Hz) to 25.
5. Click the OK button to save the configuration and close the Configure Filter dialog box.
6. Display the front panel.
7. Click the Unfiltered Signal waveform graph indicator and press the <Ctrl> key while you drag with the Positioning tool to create an additional waveform graph indicator.
8. Triple-click the Unfiltered Signal 2 label above the new waveform graph indicator and enter Filtered Signal to change the label of the indicator. You also can change the label on the Appearance page of the Graph Properties dialog box.
9. On the block diagram, wire the Result output of the Formula Express VI to the Signal input of the Filter Express VI and to the Unfiltered Signal waveform graph indicator.
10. Wire the Filtered Signal output of the Filter Express VI to the Signals input of the Amplitude and Level Measurements Express VI and to the input of the Filtered Signal waveform graph indicator.
10. Enter -2.5 in the Minimum text box and 2.5 in the Maximum text box.

11. Click the OK button to save the configuration and close the Graph Properties dialog box.

12. Repeat steps 1–11 to configure the Filtered Signal graph indicator. The x-axis and the y-axis on the Unfiltered Signal and Filtered Signal graph indicators change to reflect the new configuration.

Analyzing the Amplitude of a Signal

You can use the Amplitude and Level Measurements Express VI 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, double-click the Amplitude and Level Measurements 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. Place a checkmark in the Peak to peak checkbox. Peak to peak appears in the Results section with the corresponding value of the measurement.

4. 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 output, shown at left. You will use the Peak to Peak output in a later exercise.

Controlling the Speed of Execution

To plot the points on the waveform graphs more slowly, you can add a time delay to the block diagram. A time delay slows the speed at which a VI runs.

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.

2. Place the Time Delay Express VI inside the While Loop. The Configure Time Delay dialog box appears.

3. Enter 1.000 in the Time delay (seconds) text box and click the OK button.

11. Select File»Save. The block diagram of the Analysis VI should appear similar to Figure 3-3.

![Figure 3-3. Block Diagram of the Analysis VI](image)

Modifying the Appearance of Graphs

You can use the Display Format page of the Graph Properties dialog box to specify how the scales of the x-axis and y-axis appear on the graph.

Complete the following steps to change the format of the x-axis and y-axis of the Unfiltered Signal and Filtered Signal graphs.

1. In the front panel window, right-click the Unfiltered Signal graph indicator and select Properties from the shortcut menu. The Graph Properties dialog box appears.

2. On the Display Format page, select Time (X-Axis) from the top pull-down menu.

3. Select the Default editing mode option.

4. In the Type list, select Automatic formatting.

5. In the Digits field, enter 6 and select Significant digits from the Precision Type pull-down menu.

6. Place a checkmark in the Hide trailing zeros checkbox.

7. Select Amplitude (Y-Axis) from the top pull-down menu and repeat steps 3–6 so the y-axis configuration matches the x-axis configuration.


9. Remove the checkmark from the Autoscale checkbox.
Chapter 3  Analyzing and Saving a Signal

4. Display the front panel and run the VI. The VI runs more slowly. The loop iterates once every second.

5. Stop the VI.

Another way to control the speed of the VI is to alter the rate of data acquisition. On the block diagram, double click the Simulate Signal Express VI to display the Configure Simulate Signal dialog box. Locate the Timing section in the dialog box. The Timing section contains a number of ways to alter the rate of data acquisition and the speed at which a VI runs.

For example, one of the default settings of the VI is Simulate Acquisition Timing. This means that the VI mimics the acquisition rate of a hardware device. You can select Run as fast as possible to display data more quickly. In the Samples per second (Hz) text box, the default value is 1000, while the default value in the Number of Samples text box is 100. This means that the VI will output 100 data points spanning 0.1 second. You can change these values to change the amount of data the VI displays, as well as the rate at which the VI displays the data.

Adding a Warning Light

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

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

1. Display the Controls palette by right-clicking any blank space in the front panel window.
2. On the Express palette, select the LEDs palette.
3. Select the round LED indicator and add it to the front panel to the left of the waveform graphs.
4. Double-click the Boolean label above the LED and enter Warning to change the label of the LED.

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 options 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.
3. Display the front panel.
   A numeric indicator labeled **Peak to Peak** appears in the front panel window. This indicator displays the peak-to-peak value of the signal.

4. Run the VI.
   When the peak-to-peak value exceeds 2.0, the **Warning** indicator lights.

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

6. 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. Search for the **Write To Measurement File Express VI** and add it to the block diagram below and to the right of the **Amplitude and Level Measurements Express VI**.

   The **Configure Write To Measurement File** dialog box appears.

   The **Filename** 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 saves 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 **Filename** text box to access the `test.lvm` file.

2. In the **Configure Write to Measurement File** dialog box, locate the **If a file already exists** section and select the **Append to file** option to write 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.

### Warning the User

After you specify the values at which you want the LED to light, you must wire the LED 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. In the block diagram window, move the **Warning** terminal to the right of the **Greater Express VI**. Make sure the **Warning** terminal is inside the **While Loop**, as shown in Figure 3-4.

![Figure 3-4. Block Diagram of the Warning Light VI](image)
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. In the block diagram window, 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 Filename 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.
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 replace the Comment input with 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.

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 front panel STOP button.
5. To view the data you saved, open the test.lvm file in the LabVIEW Data directory with a spreadsheet or text-editing application. The file has a header that contains information about the Express VI.
6. Close the file after you finish looking at it and return to the Save Data VI.

Adding a Button That Stores Data When Clicked

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. Display the front panel and search the Controls palette for a rocker button. Select one of the rocker buttons and place it to the right of the waveform graphs.
2. Right-click the rocker button and select Properties from the shortcut menu to display the Boolean Properties dialog box.
3. Change the label of the button to Write to File.
4. On the Operation page of the Boolean Properties dialog box, select Latch when pressed from the Button behavior list.
5. Click the OK button to save the current configuration and close the Boolean Properties dialog box.
6. Save the VI.
Summary

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

Controls and Indicators

You can configure front panel controls and indicators 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.

Filtering Data

The Filter Express VI processes signals through filters and windows. You can use the Filter Express VI to remove noise from a signal.

Saving Data

The Write To Measurement File Express VI saves data that a VI generates and analyzes to a .lvm, .tdm, or .tdms measurement file. The text-based measurement file (.lvm) 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 .lvm file. In addition to the data an Express VI generates, the .lvm 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, take up less space on disk, and perform faster than text-based measurement files (.lvm). The TDM Streaming file (.tdms) is a binary file that provides faster writing performance than the .tdm file format and allows a simpler interface for defining properties.

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 writing data to and reading data from .lvm and .tdm files.

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.
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. Close the file after you finish looking at it.
5. Save and close the VI.
Acquiring a Signal in NI-DAQmx

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 with LabVIEW—Getting Started with DAQ—Taking an NI-DAQmx Measurement in LabVIEW book on the Contents tab in the LabVIEW Help for information about additional ways to create NI-DAQmx tasks.

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.

Creating an NI-DAQmx Task

In NI-DAQmx, a task is a collection of one or more channels, which contains 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, blank VI.
2. On the block diagram, display the Functions palette and select Express»Input to display the Input 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 Express Task dialog box appears.
4. Click Acquire Signals»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 device 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, that displays options for configuring the channel you selected to complete a task.

7. In the DAQ Assistant dialog box select the **Configuration** tab and locate the **Voltage Input Setup** section.

8. Locate the **Settings** tab. In the **Signal Input Range** section enter 10 for the **Max** value and enter -10 for the **Min** value.
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. In the block diagram window, double-click the DAQ Assistant Express VI to open the DAQ Assistant.
2. Click the Add Channels button, shown at left, and select Voltage 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. In the Timing Settings section of the Configuration page, select Continuous Samples from the Acquisition Mode pull-down menu.
   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 on the block diagram. A stop button appears 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.

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 plot legend at the top of the waveform graph.
3. On the block diagram, 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.
4. In the New Name text box, enter First Voltage Reading, and click the OK button.
5. In the DAQ Assistant dialog box, click the OK button to save the current configuration and close the DAQ Assistant.
6. Display the front panel and run the VI.
 First Voltage Reading appears in the waveform graph plot legend.
7. Save the VI.
Getting Started with the Instrument Driver Finder

Use the NI Instrument Driver Finder to search for and install LabVIEW Plug and Play instrument drivers without leaving the LabVIEW development environment.

Tip You also can visit the NI Instrument Driver Network at ni.com/idnet to find a driver for an instrument, request a driver for an instrument, and read helpful articles and tutorials about using instrument drivers.

Complete the following steps to launch and configure the NI Instrument Driver Finder.
1. From the Getting Started window, select Find Instrument Drivers to launch the Instrument Driver Finder. You also can launch the Instrument Driver Finder by selecting Help»Find Instrument Drivers or Tools»Instrumentation»Find Instrument Drivers.
2. Click the Login button to sign-in using your NI.com profile. If you do not have an ni.com profile, skip to step 4.
3. If you already have an ni.com profile, enter your email address and password and click the Login button.
4. If you do not have an ni.com profile, select the No, I need to create a profile option and click the Create Profile button. This action launches a browser window where you can create an ni.com profile. After you create a profile, return to the Instrument Driver Finder window and login with your new information.

You are now ready to search for, install, and use instrument drivers with the Instrument Driver Finder.

Finding and Installing Instrument Drivers

Complete the following steps to search for and install an instrument driver using the Instrument Driver Finder.
1. On the Configure Search page, click the Scan for Instruments button. This action prompts the NI Instrument Driver Finder to search for connected instruments. All results display under the Connected Instruments folder in the left-hand sidebar. If you do not currently have an instrument connected, the Instrument Driver Finder will return a result that reads <no connected instrument detected>.
2. Expand the Connected Instruments folder to display the search results and select an instrument from the list.

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 of the front panel waveform graph.
1. Expand the plot legend of the waveform graph to display two plots.
2. Run the VI.
Two plots appear on the graph and the plot legend displays both plot names.
3. Click the icon that is to the right of 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.
4. Change the plot color of Second Voltage Reading.
5. Stop the VI.
6. Save the VI.
7. Close the VI. The Getting Started window opens.

Communicating with an Instrument: Using Instrument Drivers and the Instrument I/O Assistant

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. Instrument drivers simplify instrument control and reduce test program development time by eliminating the need to learn the programming protocol for each instrument. Use an instrument driver for instrument control when possible. National Instruments provides thousands of instrument drivers for a wide variety of instruments.

In the following exercises, you will use instrument drivers and the Instrument I/O Assistant to communicate with an instrument. You must have instrument installed to fully complete the following exercises.

Note These exercises refer to traditional, third-party instruments. Refer to ni.com/modularinstruments for more information about communicating with NI modular instruments.
3. Select a manufacturer from the Manufacturer pull-down menu, and enter any keywords in the Additional Keywords section. Then click the Search button. A list of available instrument drivers appears on the Search Results page. The driver result for the most recent version of LabVIEW appears first in the list.

If your search does not return any results, a sidebar with search tips appears. Refer to the IDNet Web site at ni.com/idnet for more information about searching for instrument drivers.

Note The NI Instrument Driver Finder displays drivers only for LabVIEW 7.0 or later. If you need to download an older version of an instrument driver, go to the IDNet Web site at ni.com/idnet to search for and download instrument drivers compatible with older versions of LabVIEW.

4. Select the driver you want to install and then click the Install button.

After the driver downloads successfully, the Download Success window displays. This window contains options for exploring and using the new driver. After downloading, the new driver also appears in the Configure Search page under the Installed Instrument Driver folder.

5. Close the Download Success window to return to the Configure Search window.

Note You also can create your own instrument drivers. Refer to the Controlling Instruments—Using Instrument Drivers book on the Contents tab in the LabVIEW Help for more information about creating instrument drivers.

Selecting an Instrument Using the Instrument I/O Assistant

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 installed to use the Instrument I/O Assistant Express VI. You install the Instrument I/O Assistant from the National Instruments Device Drivers CD.

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. Open a new VI and display the front panel window.
3. From the Input palette, select the Instrument I/O Assistant Express VI and add it to the block diagram. The Instrument I/O Assistant dialog box appears.
4. If the help is not visible to the right of the dialog box, 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.
5. 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.

Note Not all drivers have all options available. For example, if a driver does not have a project file, the Open Project button appears dimmed. Refer to the IDNet Web site at ni.com/idnet for more information about all instrument drivers.

Using Instrument Drivers

After installing an instrument driver, you can explore example programs to learn more about using the instrument driver.

1. Double-click the newly installed instrument driver in the Installed Instrument Driver folder to display the Start Using Instrument Driver page.
2. The Start Using Instrument Driver page allows you to explore and customize the new driver. The following are recommendations to help you start using the new driver.

   • To view the new driver in the Project Explorer window, click the Open Project button. In the Project Explorer window, you can explore the VIs, folders, and supplemental files that make up the...
Wiring a Command to an Instrument

After you acquire data from the instrument, you can add an input parameter to an instrument command. The parameter becomes an input to the VI or function.

1. If necessary, configure the properties of the instrument.
2. 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.

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.
3. Highlight the command in the Enter a command text box and click the Add parameter button to add a parameter to the command.
4. Enter a default value for the parameter in the Test value text box.
5. Enter a name for the parameter in the Parameter name text box. You use this name to reference the parameter in the application.
6. Click the OK button to save the current configuration and close the Instrument I/O Assistant dialog box.

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.

Add the DAQ Assistant Express VI to 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 with LabVIEW Getting Started with DAQ Taking an NI-DAQmx Measurement in LabVIEW book on the Contents tab in the LabVIEW Help for information about the DAQ Assistant.

6. If necessary, configure the properties of the instrument.
7. 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.

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.
3. Click the Run this step 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.

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.
Instrument I/O Assistant Express VI

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 adding the Instrument I/O Assistant Express VI to 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.

Instrument Drivers

Use the NI Instrument Driver Finder to search for and install LabVIEW Plug and Play instrument drivers without leaving the LabVIEW development environment.

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.

Refer to the Controlling Instruments»Using Instrument Drivers book on the Contents tab in the LabVIEW Help for more information about the Instrument Driver Finder.

You also can visit the NI Instrument Driver Network at ni.com/idnet to find a driver for an instrument, or you can create your own instrument drivers. Refer to the Controlling Instruments»Using Instrument Drivers book on the Contents tab in the LabVIEW Help for more information about creating instrument drivers.

Tasks in NI-DAQmx

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

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.
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. On other subpalettes you can find all the controls and indicators that you can use to create the front panel. 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 Change Visible Categories from the shortcut menu to display the Change Visible Categories dialog box. Then place checkmarks in the checkboxes next to the categories you want to view on the Controls palette.

All VIs and Functions

The Express VIs and structures located on the Express subpalette of the Functions palette are a small 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 Change Visible Categories from the shortcut menu to display the Change Visible Categories dialog box. Then place checkmarks in the checkboxes next to the categories you want to view 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

You can use an existing VI or a VI you create as a subVI. 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 the default icon or create a custom icon using the Icon Editor.

Refer to the Fundamentals»Creating VIs and SubVIs book on the Contents tab in the LabVIEW Help for more information about creating VIs, configuring them as subVIs, and creating icons.

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.
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.

- **Share data within an application or across a network**—You can create configured software items called shared variables to share data among VIs or between locations on the block diagram that you cannot connect with wires. 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.

Functions

Functions are the fundamental operating elements of LabVIEW. Unlike VIs, functions do not have front panels or block diagrams. Functions provide the basic building blocks for programming a VI, interfacing with hardware and software, and performing other essential tasks in LabVIEW. Refer to the Fundamentals>Building the Block Diagram book on the Contents tab in the LabVIEW Help for more information about functions.

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.

Express VIs generate and acquire data using the dynamic data type. 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 data types.
Chapter 5  Using Other LabVIEW 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 files not specific to LabVIEW, 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, JDA, Touch Panel, DSP, or embedded target. Refer to the specific module documentation for more information about using projects with these targets. Refer to the Fundamentals»Working with Projects and Targets book on the Contents tab in the LabVIEW Help for more information about using LabVIEW projects.

- **Access other Windows applications**—You can use LabVIEW as a .NET or ActiveX client to access the objects, properties, methods, and events associated with .NET server or ActiveX applications. Refer to the Fundamentals»Windows Connectivity book on the Contents tab in the LabVIEW Help for more information about these features.

- **Write mathematical formulas, equations, and scripts**—You can use various nodes to perform mathematical operations on the block diagram. You also can use the LabVIEW MathScript text-based language to write mathematical functions and scripts. Refer to the Fundamentals»Formulas and Equations book on the Contents tab in the LabVIEW Help for more information about these features.

---

Technical Support and Professional Services

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

- **Support**—Technical support at ni.com/support includes the following resources:
  - **Self-Help Technical Resources**—For answers and solutions, visit ni.com/support for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on. Registered users also receive access to the NI Discussion Forums at ni.com/forums. NI Applications Engineers make sure every question submitted online receives an answer.
  - **Standard Service Program Membership**—This program entitles members to direct access to NI Applications Engineers via phone and email for one-to-one technical support as well as exclusive access to on demand training modules via the Services Resource Center. NI offers complementary membership for a full year after purchase, after which you may renew to continue your benefits.

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

- **Training and Certification**—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.

- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>automatic scaling</td>
<td>Ability of scales to adjust to the range of plotted values. On graph scales, autoscaling determines maximum and minimum scale values.</td>
</tr>
<tr>
<td>block diagram</td>
<td>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.</td>
</tr>
<tr>
<td>Boolean controls and indicators</td>
<td>Front panel objects to manipulate and display Boolean (TRUE or FALSE) data.</td>
</tr>
<tr>
<td>broken Run button</td>
<td>Button that replaces the Run button when a VI cannot run because of errors.</td>
</tr>
<tr>
<td>broken VI</td>
<td>VI that cannot run because of errors; signified by a broken arrow in the broken Run button.</td>
</tr>
</tbody>
</table>
**channel**

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. A counter also can be a physical channel, although the counter name is not the name of the terminal where the counter measures or generates the digital signal.

2. Virtual—A collection of property settings that can include a name, a physical channel, input terminal connections, the type of measurement or generation, and scaling information. You can define NI-DAQmx virtual channels outside a task (global) or inside a task (local). Configuring virtual channels is optional in Traditional NI-DAQ (Legacy) and earlier versions, but is integral to every measurement you take in NI-DAQmx. In Traditional NI-DAQ (Legacy), you configure virtual channels in MAX. In NI-DAQmx, you can configure virtual channels either in MAX or in your program, and you can configure channels as part of a task or separately.

3. Switch—A switch channel represents any connection point on a switch. It 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.

**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.

**data acquisition (DAQ)**

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. See also 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.

**dynamic data type**

Data type used by Express VIs that includes the data associated with a signal and attributes that provide information about the signal, such as the name of the signal or the date and time LabVIEW acquired the data. Attributes specify how the signal appears on a graph or chart.
<table>
<thead>
<tr>
<th>Glossary</th>
<th>Glossary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I</strong></td>
<td><strong>E</strong></td>
</tr>
<tr>
<td>I/O</td>
<td>Error list window</td>
</tr>
<tr>
<td>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.</td>
<td>Window that displays errors and warnings occurring in a VI and in some cases recommends how to correct the errors.</td>
</tr>
<tr>
<td>icon</td>
<td>error message</td>
</tr>
<tr>
<td>Graphical representation of a node on a block diagram.</td>
<td>Indication of a software or hardware malfunction or of an unacceptable data entry attempt.</td>
</tr>
<tr>
<td>indicator</td>
<td>Express VI</td>
</tr>
<tr>
<td>Front panel object that displays output, such as a graph or LED.</td>
<td>A subVI designed to aid in common measurement tasks. You configure an Express VI using a configuration dialog box.</td>
</tr>
<tr>
<td>instrument driver</td>
<td>For Loop</td>
</tr>
<tr>
<td>A set of high-level functions that control and communicate with instrument hardware in a system.</td>
<td>Iterative loop structure that executes its subdiagram a set number of times. Equivalent to text-based code: For i = 0 to n - 1, do....</td>
</tr>
<tr>
<td>Instrument I/O Assistant</td>
<td>front panel</td>
</tr>
<tr>
<td>Add-on launched from the Instrument I/O Assistant Express VI that communicates with message-based instruments and graphically parses the response.</td>
<td>Interactive user interface of a VI. Front panel appearance imitates physical instruments, such as oscilloscopes and multimeters.</td>
</tr>
<tr>
<td><strong>L</strong></td>
<td>function</td>
</tr>
<tr>
<td>label</td>
<td>Built-in execution element, comparable to an operator, function, or statement in a text-based programming language.</td>
</tr>
<tr>
<td>Text object used to name or describe objects or regions on the front panel or block diagram.</td>
<td><strong>Functions</strong> palette</td>
</tr>
<tr>
<td>LabVIEW</td>
<td>Palette that contains VIs, functions, block diagram structures, and constants.</td>
</tr>
<tr>
<td>Laboratory Virtual Instrument Engineering Workbench. LabVIEW is a graphical programming language that uses icons instead of lines of text to create programs.</td>
<td><strong>G</strong></td>
</tr>
<tr>
<td>LED</td>
<td>General Purpose Interface Bus</td>
</tr>
<tr>
<td>legend</td>
<td>graph</td>
</tr>
<tr>
<td>Object a graph or chart owns to display the names and plot styles of plots on that graph or chart.</td>
<td>2D display of one or more plots. A graph receives and plots data as a block.</td>
</tr>
<tr>
<td>MAX</td>
<td>Measurement &amp; Automation Explorer</td>
</tr>
</tbody>
</table>
measurement device

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.

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.

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.
V

VI  See 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.

glossary

string  Representation of a value as text.

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

subpalette  Palette that you access from another palette that is above the subpalette in hierarchy.

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

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.

template VI  VI that contains common controls and indicators from which you can build multiple VIs that perform similar functions. Access template VIs from the New dialog box.

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.
Index

A
acquiring information for instruments, 4-11
signals, 4-2
Acquiring a Signal VI block diagram (figure), 1-14
adding
See also creating channels to tasks, 4-6
to the block diagram, 2-5
to the front panel, 1-6
Arithmetic & Comparison palette, 1-12

B
blank VIs, 2-2
block diagram, 1-6, 1-19
customizing, 2-6
diagnostic tools (NI resources), A-1
displaying
data from DAQ devices, 4-5
data in tables, 2-9, 2-14
diagnosis dialog boxes, 1-20
diagnostic tools (NI resources), A-1
dynamic data, converting from and to, 5-3
Error list window, 2-8, 2-14
eyes, 2-14
errors, 2-14
equivalence, 5-1
Expression palette, 1-6
expressions, 5-1
Expressions palette, 1-6
Expressions to VIs, 2-14
Expressions VIs, 2-14
Express VIs, 1-20
Amplitude and Level Measurements, 3-3, 3-9
Build Table, 2-9

C
calling code from text-based languages, 5-4, 5-5
to text-based languages, 5-4, 5-5
changing signal types, 1-8
to signal types, 1-8
channels, 4-2
to channels, 4-2
adding to tasks, 4-6
renaming, 4-5
adding to tasks, 4-6
renaming, 4-5

D
data
displaying from DAQ devices, 4-5
in tables, 2-9, 2-14
graphing from DAQ devices, 4-5
DAQ Assistant Express VI, 4-2, 4-12
DAQ devices, 4-2
deleting wires, 1-11
deselecting objects, 1-10
deserialize, 5-3
diagnostic tools (NI resources), A-1
 drivers, instrument, 4-7
 NI resources, A-1
dynamic data, converting from and to, 5-3

E
Error list window, 2-8, 2-14
eyes, 2-14
displaying in Context Help window, 2-14
list, 2-8, 2-14
window, 2-8, 2-14
example VIs
NI Example Finder, 2-11
examples (NI resources), A-1
Execution Control palette, 2-6
execution speed, controlling, 2-9
Express VIs, 1-20
Amplitude and Level Measurements, 3-3, 3-9
Build Table, 2-9

getting
promoted to a user, 3-15

I
Index

ni.com

© National Instruments Corporation
Index

to files, 3-13, 3-14, 3-17
when prompted by user, 3-15, 3-16
Scaling and Mapping Express VI
defining slope, 1-12
searching
effects, 2-11
help, 2-4, 2-11, 2-13
palettes, 2-6
selecting
instruments, 4-10
objects, 1-10
shared libraries, building, 5-5
signals
acquiring, 4-2
analyzing, 3-9
changing type, 1-8
graphing, 1-15
modifying, 1-11, 2-3
Simulate Signal Express VI, 1-8
simulated signals, customizing, 3-4
software (NI resources), A-1
subVIs, 5-2
system requirements, x

T
tables, 2-9
displaying data, 2-14
tasks
adding new channels, 4-6
NI-DAQmx, 4-13
TDM, See .tdm files
.tdm files, 3-17
.tdms files, 3-17
technical support, A-1
template VIs, 1-3, 1-4, 1-18
text-based languages, calling code, 5-4, 5-5
Time Delay Express VI, 2-9
tools
Operating, 1-11
Positioning, 1-10

Wiring, 1-10
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
blank, 2-2
building, 1-2
customizing menus, 5-5
icons, 5-2
new, 2-2
programmatically controlling, 5-4
publishing on the Web, 5-4
running, 1-11
continuously, 2-6
subVIs, 5-2
template, 1-3, 1-4, 1-18
voltage, analyzing, 3-9

W
Warning Light VI block diagram (figure), 3-12
warning lights, adding, 3-10
Web resources, A-1
While Loop, 2-7
wires
broken, 2-8, 2-14
deleting, 1-11
wiring
objects on the block diagram, 1-10
tool, 1-10
Write to Measurement File Express VI, 3-14,
3-17
saving data, 3-14