

LabVIEW[™]

Getting Started with LabVIEW and Tektronix Open Windows Oscilloscopes

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

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

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 3262 3599,
Canada (Calgary) 403 274 9391, Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949,
Canada (Québec) 514 694 8521, Canada (Toronto) 905 785 0085, China 86 21 6555 7838,
Czech Republic 02 2423 5774, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24,
Germany 089 741 31 30, Greece 01 42 96 427, Hong Kong 2645 3186, India 91 80 4190000,
Israel 03 6393737, Italy 02 413091, Japan 03 5472 2970, Korea 02 3451 3400, Malaysia 603 9596711,
Mexico 001 800 010 0793, Netherlands 0348 433466, New Zealand 09 914 0488, Norway 32 27 73 00,
Poland 22 3390 150, Portugal 210 311 210, Russia 095 238 7139, Singapore 65 6 226 5886,
Slovenia 3 425 4200, South Africa 11 805 8197, Spain 91 640 0085, Sweden 08 587 895 00,
Switzerland 056 200 51 51, Taiwan 02 2528 7227, United Kingdom 01635 523545

For further support information, see the *Technical Support and Professional Services* appendix. To comment on the documentation, send email to techpubs@ni.com.

© 2002 National Instruments Corporation. All rights reserved.

Important Information

Warranty

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

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

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

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THEREFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

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

Trademarks

LabVIEW™, National Instruments™, NI™, ni.com™, and NI-VISA™ are trademarks of National Instruments Corporation.

Tektronix and Tek are registered trademarks of Tektronix, Inc. Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Patents

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

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Contents

About This Manual

Conventions	x
-------------------	---

Chapter 1

Introduction to LabVIEW

What Is LabVIEW?	1-1
How Does LabVIEW Work?	1-2
Front Panel.....	1-2
Block Diagram.....	1-2
Data Flow	1-3
Palettes.....	1-3
Tools Palette.....	1-3
Controls Palette	1-4
Functions Palette	1-5
Navigating the Controls and Functions Palettes	1-6
Using LabVIEW with Open Windows Oscilloscopes	1-6
Components of the LabVIEW Add-On for Tektronix	
Open Windows Oscilloscopes	1-7
LabVIEW Scope Analysis Demo	1-7
Example VIs.....	1-7
Example Browser	1-8
Add-On Palettes for Tektronix Scopes	1-8
Where Do I Start?	1-9
LabVIEW Tutorial.....	1-9

Chapter 2

LabVIEW Scope Analysis

Using the Scope Analysis Demo	2-1
Specifying a Signal Source.....	2-1
Selecting Analysis and Signal Processing	2-2
Examining Results, Exporting Data, and Generating Reports	2-2
Run the Scope Analysis Demo	2-3

Chapter 3

Virtual Instruments

Search for Examples.....	3-1
Search for Scope-Specific Examples	3-1
Finding General LabVIEW Examples	3-3
Build a Virtual Instrument.....	3-3
Create a User Interface.....	3-3
Build the Block Diagram	3-4
Wire and Run the VI.....	3-6
Acquire and Display Waveform Data.....	3-7

Chapter 4

Scope-Specific Programming

Using Scope-Specific VIs.....	4-1
Options for LabVIEW Communication with a Scope	4-1
Programming with Scope-Specific VIs.....	4-1
Commonly Used Scope-Specific VIs	4-2
Initialize and Close	4-2
Get and Set Waveform	4-3
Additional Scope-Specific Programming Information	4-3

Chapter 5

Debugging

Use Execution Highlighting	5-1
Single-Step with Probes	5-2

Chapter 6

Where to Go from Here

Online Help	6-1
Web Resources	6-1
National Instruments' Commitment to You	6-2
Customer Education	6-2
Alliance Program	6-2

Appendix A
System Requirements

Appendix B
Technical Support and Professional Services

Glossary

About This Manual

This manual contains basic information you need to use LabVIEW with a Tektronix Open Windows oscilloscope.

Use the following questions and answers to determine where you should begin in this manual.

What is LabVIEW?

Chapter 1, *Introduction to LabVIEW*, presents the basics of LabVIEW, describes how VIs work, and shows you how to navigate the LabVIEW environment.

How can I use LabVIEW with my scope?

Chapter 2, *LabVIEW Scope Analysis*, demonstrates how to work with the LabVIEW Scope Analysis Demo, a stand-alone application included with the LabVIEW Add-On for Tektronix Open Windows Oscilloscopes. The application demonstrates analysis, signal processing, report generation, and other components from the LabVIEW Full Development System.

Chapter 3, *Virtual Instruments*, contains activities that demonstrate how you can create VIs that work with your scope. The chapter also contains information about the LabVIEW Add-On for Tektronix Open Windows Oscilloscopes, a set of software tools that you can use with LabVIEW and your scope.

Chapter 4, *Scope-Specific Programming*, contains methods for programming your scope using LabVIEW. The chapter contains an overview of the options available for communication between custom software applications and LabVIEW, information about working with the palette of VIs from the LabVIEW Scope Add-On, and instructions on how to transfer a waveform from the scope to a custom VI.

Why should I be interested in using LabVIEW with my scope?

You can use LabVIEW to rapidly customize your scope to perform a variety of tasks, including the following:

- Customize analysis and signal processing of acquired signals, either online (as signals are acquired) or offline (loaded from a file).
- Automate scope-related tasks and sequences of tasks.
- Expand user interface options.

- Create reports with standard or custom file formats and content.
- Publish reports live over the Internet.
- Control a scope remotely on the Web.

Where can I go for help using LabVIEW with my scope?

Chapter 6, *Where to Go from Here*, shows you how to find and use resources available for help with LabVIEW programming.

Appendix B, *Technical Support and Professional Services*, lists general National Instruments support resources.

The LabVIEW/Tektronix resources page at ni.com/tek provides support resources for using LabVIEW with Tektronix Open Windows oscilloscopes.

Conventions

This manual uses the following conventions:

»

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



This icon denotes a tip, which alerts you to advisory information.



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

bold

Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes palette and parameter names.

italic

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

monospace

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

monospace italic

Italic text in this font denotes text that is a placeholder for a word or value that you must supply. For example, in the term **TDS#k**, substitute the scope series number for the # symbol.

Introduction to LabVIEW

LabVIEW expands the capability of an Open Windows oscilloscope so you can develop and run custom Windows-based applications that complement the functionality of the scope and fulfill the specifics of your application.

What Is LabVIEW?

LabVIEW is a graphical development environment that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabVIEW uses dataflow programming, where the flow of data determines execution.

In LabVIEW, you build a user interface by using a set of tools and objects. The user interface is known as the front panel. You then add code using graphical representations of functions to control the front panel objects. The block diagram contains this code. In some ways, the block diagram resembles a flowchart.

LabVIEW is integrated fully for communication with hardware such as GPIB, VXI, PXI, RS-232, RS-485, and data acquisition control, vision, and motion control devices. LabVIEW also has built-in features for connecting your application to the Web using the LabVIEW Web Server and software standards such as TCP/IP networking and ActiveX.

Using LabVIEW, you can create 32-bit compiled applications that give you the fast execution speeds needed for custom data acquisition, test, measurement, and control solutions. You also can create stand-alone applications and shared libraries, like DLLs, because LabVIEW is a true 32-bit compiler.

LabVIEW contains comprehensive libraries for data collection, analysis, presentation, and storage. LabVIEW also includes traditional program development tools. You can set breakpoints, animate program execution, and single-step through the block diagram to make debugging and development easier.

LabVIEW also provides numerous mechanisms for connecting to external code or software through DLLs, shared libraries, ActiveX, and so on. In addition, numerous add-on tools are available for a variety of application needs. You can purchase several add-on software toolsets for developing specialized applications. All the toolsets integrate seamlessly in LabVIEW. Refer to the National Instruments Web site at ni.com/labview for more information about these toolsets.

How Does LabVIEW Work?

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. Every VI uses functions that manipulate input from the user interface or other sources and display that information or move it to other files or other computers.

A VI contains the following three components:

- **Front panel**—Serves as the user interface.
- **Block diagram**—Contains the graphical source code of the VI that defines its functionality.
- **Icon and connector pane**—Identifies the VI so you can use the VI in another VI. A VI within another VI is called a subVI. A subVI corresponds to a subroutine in text-based programming languages.

Front Panel

The front panel is the user interface of the VI. You build the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates.

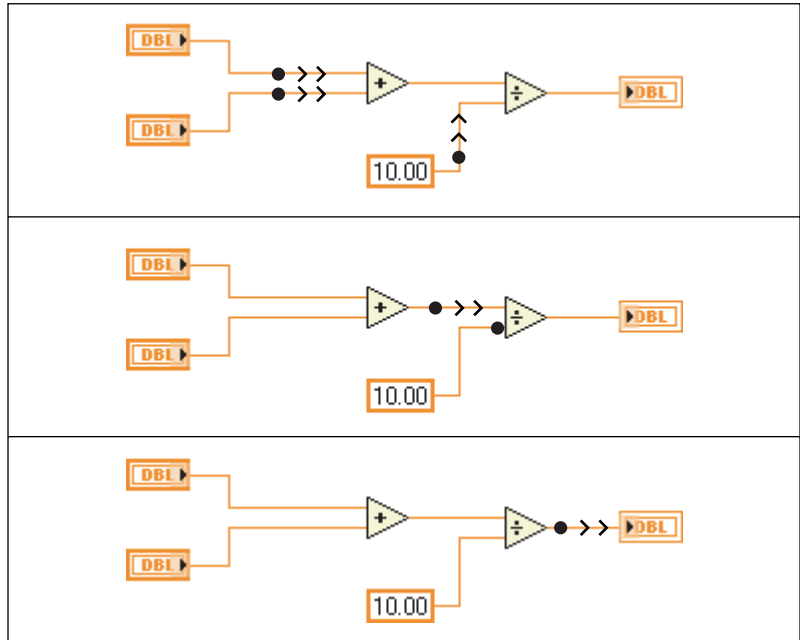
Block Diagram

After you build the front panel, you add code using graphical representations of functions to control the front panel objects. The block diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram.

Wires connect each of the nodes on the block diagram, including control and indicator terminals, subVIs, functions, and structures.

Data Flow

LabVIEW follows a dataflow model for running VIs. A block diagram node executes when all its inputs are available. When a node completes execution, it supplies data to its output terminals and passes the output data to the next node in the dataflow path.



Palettes

LabVIEW palettes contain the objects you need to create and edit the front panel and block diagram.

Tools Palette

The **Tools** palette is available on the front panel and the block diagram. A tool is a special operating mode of the mouse cursor. When you select a tool, the cursor icon changes to the tool icon. Use the tools to operate and modify front panel and block diagram objects.

Select **Window»Show Tools Palette** to display the **Tools** palette. You can place the **Tools** palette anywhere on the screen.



Controls Palette

The **Controls** palette is available only on the front panel. The **Controls** palette contains the controls and indicators you use to create the front panel. Select **Window»Show Controls Palette** or right-click the front panel workspace to display the **Controls** palette. You can place the **Controls** palette anywhere on the screen.



Functions Palette

The **Functions** palette is available only on the block diagram. The **Functions** palette contains the VIs and functions you use to build the block diagram. Select **Window»Show Functions Palette** or right-click the block diagram workspace to display the **Functions** palette. You can place the **Functions** palette anywhere on the screen.



Navigating the Controls and Functions Palettes

Use the navigation buttons on the **Controls** and **Functions** palettes to navigate and search for controls, VIs, and functions. When you click a subpalette icon, the entire palette changes to the subpalette you selected. You also can right-click a VI icon on the palette and select **Open VI** from the shortcut menu to open the VI.

The **Controls** and **Functions** palettes contain the following navigation buttons:



- **Up**—Takes you up one level in the palette hierarchy.
- **Search**—Changes the palette to search mode. In search mode, you can perform text-based searches to locate controls, VIs, or functions in the palettes.
- **Options**—Displays the **Function Browser Options** dialog box, from which you can configure the appearance of the palettes.

Using LabVIEW with Open Windows Oscilloscopes

Open Windows oscilloscopes from the TDS 5000, 6000, 7000, and 8000 series are Windows-based and contain scope-specific software for acquisition, connectivity, and control. By default, the scope user interface runs Windows-based application software that performs the following scope-specific tasks:

- Presents the scope user interface
- Configures scope hardware based on commands from the user and signal content
- Displays acquired signals and derivations of acquired signals

You can use LabVIEW to extend the feature list of your scope to include the following tasks:

- Custom analysis and signal processing of acquired signals
- Automation of scope-related tasks and sequences of tasks
- A unique and customizable user interface
- Custom reports, including reports that you can publish live over the Internet
- Remote, Internet-based control and monitoring

Components of the LabVIEW Add-On for Tektronix Open Windows Oscilloscopes

The LabVIEW Add-On for Tektronix Open Windows Oscilloscopes consists of four main software components: a scope analysis demo, example VIs, an example browser, and a palette of VIs and functions.

LabVIEW Scope Analysis Demo

The LabVIEW Add-On for Tektronix Open Windows Oscilloscopes includes a stand-alone application called the LabVIEW Scope Analysis Demo. The application runs independent of LabVIEW.



Note The features that the application demonstrates all are available in the LabVIEW Full Development System, except for the Application Builder, which is available in the Professional Development System. The Application Builder is necessary to compile and create an executable stand-alone application.

You can use the LabVIEW Scope Analysis Demo to perform interactive signal processing and analyze live or file-based signals from the scope.

Select **Start»Program Files»National Instruments»LabVIEW Scope Analysis Demo (TK TDS#k)**, where # represents the scope series number, to run the demo utility.

Example VIs

The LabVIEW Add-On for Tektronix Open Windows Oscilloscopes contains a set of documented example VIs that include source code. Use these examples as teaching tools and as starting points when you build custom LabVIEW applications.

The example VIs are available through the Example Browser or from within the LabVIEW development environment. By default, LabVIEW installs the example VIs into a subdirectory of the LabVIEW installation directory, as follows:

```
LabVIEW\examples\Tektronix\TK TDS#k
```

where *LabVIEW* is the main installation directory for the LabVIEW development environment and # is the number that corresponds to your scope line.

Example Browser

The Example Browser is an interactive interface that presents a categorized list of the examples related to the LabVIEW Add-On for Tektronix Open Windows Oscilloscopes. Double-click an entry on the list to open an example.

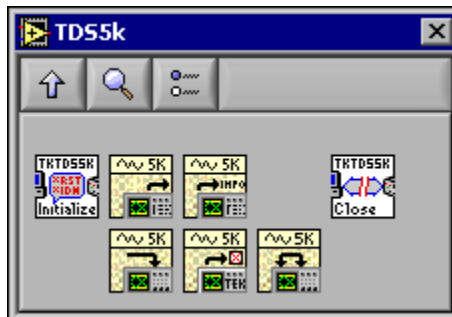
In the LabVIEW Evaluation, Full, and Professional Development Systems, you can access the Example Browser by selecting **Help»Scope Example Browser**.

If you are using the LabVIEW Add-On for Tektronix Open Windows Oscilloscopes with the Evaluation version of LabVIEW, you can use any of several methods to open the Example Browser. You can click the **Examine Scope-Specific Examples** button in the dialog box that appears when you first launch LabVIEW. You also can launch LabVIEW or, if it is open, close all open VIs to access the LabVIEW dialog box and click the **Tektronix Examples** button. You also can select **Help»Scope Example Browser**.

Add-On Palettes for Tektronix Scopes

The scope-specific palettes located on the **Functions»Tektronix** palette contain VIs that simplify waveform transfer between custom LabVIEW applications and scopes. Refer to the [Functions Palette](#) section for more information about using the **Functions** palette with the block diagram.

The **Tektronix** palette installs with the LabVIEW Add-On for Tektronix Open Windows Oscilloscopes. The TDS 5000, 6000, 7000, and 8000 scope series each have a palette containing five VIs. For example, the TDS 5000 VIs are located on the **Functions»Tektronix»TDS5k** palette, as shown in the following example.



Where Do I Start?

If you are new to LabVIEW, use this manual and the online *LabVIEW Tutorial* to help you get started quickly. The *LabVIEW Tutorial* introduces you to the LabVIEW environment. This manual teaches you how to build VIs that interact with scopes, how to find example VIs, and how to debug VIs.

If you have used LabVIEW before, but not with a scope, use the Example Browser to find scope-specific example VIs. Refer to the *Search for Scope-Specific Examples* section of Chapter 3, *Virtual Instruments*, for more information about working with the Example Browser.



You can complete the activities in this book in approximately 80 minutes.

LabVIEW Tutorial

Use the online *LabVIEW Tutorial* to learn basic LabVIEW concepts. The tutorial guides you through several activities to familiarize you with graphical programming. Access the *LabVIEW Tutorial* by selecting **Help»Contents and Index** or by clicking the **LabVIEW Tutorial** button in the **LabVIEW** dialog box, shown in the following illustration.



You can complete the *LabVIEW Tutorial* in approximately 15 minutes.



Either launch LabVIEW or close all open VIs including the Demo VI if you are already running LabVIEW to access the **LabVIEW** dialog box.

After you finish the *LabVIEW Tutorial*, continue with the activities in this manual to learn how to build LabVIEW programs that interact with oscilloscopes.

You should complete the activities in this manual in the order in which they appear because subsequent activities build on the VIs you create. Approximate completion times are given for each activity and each section within an activity.

LabVIEW Scope Analysis

This chapter teaches you how to use the LabVIEW Scope Analysis Demo, a stand-alone application that demonstrates LabVIEW features such as analysis, signal processing, and report generation. Refer to the [Run the Scope Analysis Demo](#) section to learn how to perform specific tasks using the demo.

Using the Scope Analysis Demo

The user interface of the LabVIEW Scope Analysis Demo consists of the following three components:

- The **Acquire** tab contains controls that you can use to acquire a signal.
- The **Analyze** tab contains controls that you can use to select and configure analysis and signal processing.
- The **Present** tab contains controls that you can use to examine results and create reports.

On the main screen of the LabVIEW Scope Analysis Demo, click the tabs on the bottom of the display to switch to different sections of the demo.

You can change acquisition, triggering, and other parameters on the scope in parallel with the execution of this application and other VIs you develop. The results of the changes appear in any plots, graphs, or results that are visible. Configure horizontal scaling, vertical scaling, triggering, and other scope-specific functions through the built-in software and hardware from the scope. For example, a vertical scaling knob does not appear in the application because you can change that parameter by turning a physical knob on the front of the scope.

Specifying a Signal Source

Use the controls on the **Acquire** tab to specify the signal source or sources you want to analyze. Use the **Waveform A Source** and **Waveform B Source** controls to specify one or two waveform sources. The controls allow you to select **Channel**, **Off**, or **File** options for the waveform sources you want to use. Select **Channel** to acquire live waveform signals with updates that occur as new data becomes available. Use the **Source A** and/or

Source B controls to specify the channel source. Select **File** to use a waveform from a file that you specify in the **Filename A** and/or **Filename B** listboxes.

Selecting Analysis and Signal Processing

Use the controls on the **Analyze** tab to select the type of analysis or signal processing to apply to the signal or signals you specify on the **Acquire** tab. Select the type of analysis you want from the list on the lower left corner of the **Analyze** tab. After you select the analysis type, the top of the window displays appropriate graphs and numeric indicators for any scalar values associated with the measurement. Use the group of controls that appears on the right of the list to set parameters specific to the type of analysis you select. Use **Analyze Channel** to specify which channel you want to analyze or process.

For example, if you select **Tone Detection** from the list of analysis options, the top graph becomes a waveform plot with a single time-domain trace. The trace displays the waveform on the **Export Signal** control. Three scalar measurements to the right of the graph display detected amplitude, phase, and frequency.

Examining Results, Exporting Data, and Generating Reports

Use the controls on the **Present** tab to examine, export, and report the results of the analysis that you specify on the **Analyze** tab. The **Present** tab window includes an expanded version of the graph or graphs on the **Analyze** tab and numeric indicators of scalar measurements associated with the current analysis.

Use the buttons on the bottom of the **Present** tab to generate reports. Click the **Export ASCII Data** button to generate a file with data from the graph that the demo displays. The file format is standard text with comma separators. You can import this file format into text editors, standard spreadsheet applications, and math software. Click the **Export HTML Snapshot** button to generate an HTML file with an image of the **Present** tab window in JPEG format.

If you select **None (off)** as the analysis option on the **Analyze** tab, an expanded view of an unanalyzed acquisition appears on the **Present** tab window. Place a checkmark in the **Cursors On/Off** checkbox to display cursor controls on the time-domain graph. The **Cursors** indicators display the coordinates of two cursors and the difference between the x and y values of these coordinates. Use the Positioning tool, shown at left, located at the bottom of the graph to move the cursors.



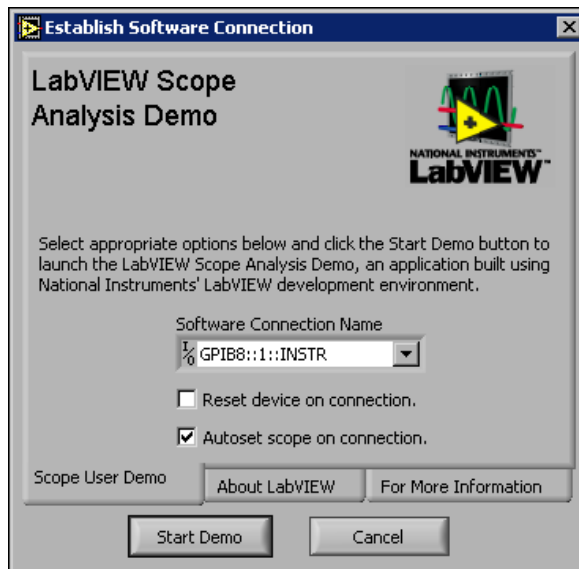
Run the Scope Analysis Demo

This activity teaches you how to use the LabVIEW Scope Analysis Demo to display and analyze signals from a scope.

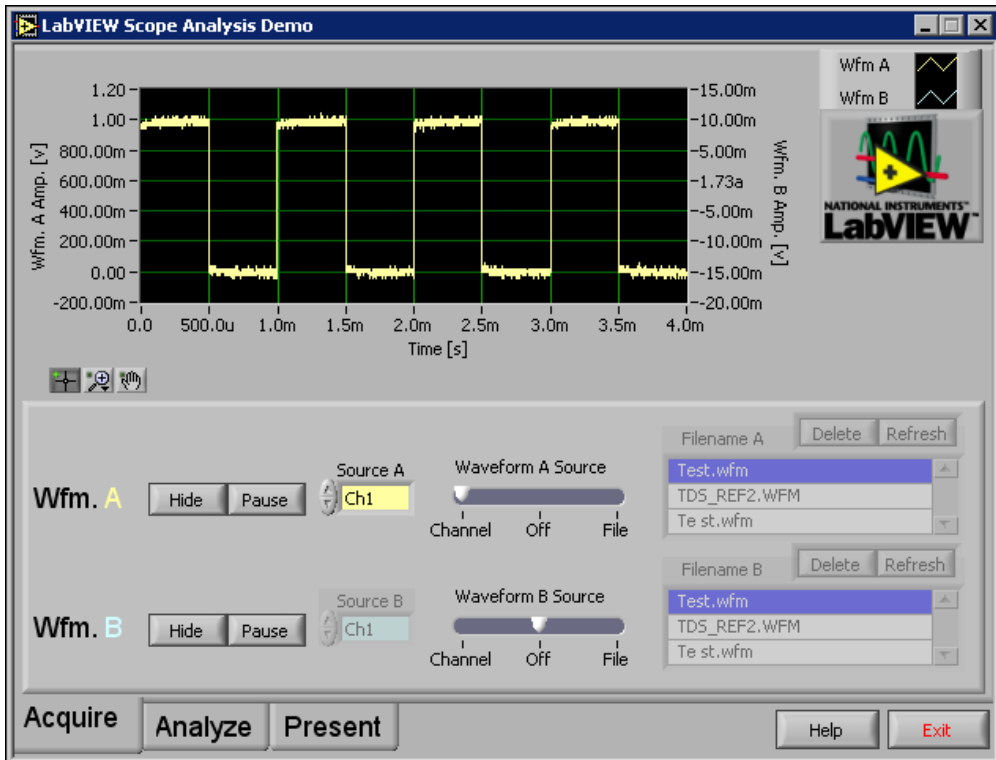


You can complete this activity in approximately 10 minutes.

1. On the scope, attach a probe to CH1 and connect it to the **Probe Comp** output.
2. Select **Start»Programs»National Instruments»LabVIEW Scope Analysis Demo (TDS #k)»LabVIEW Scope Analysis Demo (TDS #k)** to run the LabVIEW Scope Analysis Demo. The LabVIEW Scope Demo configuration screen appears.
3. The **Software Connection Name** control specifies a unique connection name that the software uses to communicate with the scope hardware. The default, **GPIB8::1::INSTR**, works if you are running the application directly on the scope and the TekScope application is running, as shown in the following example. If the entry is not available, verify that the TekScope application is running. If it is not running, click the **Cancel** button, restart the TekScope application, and return to the previous step. If you are running this application on a remote PC connected to your scope through an instrument control interface such as GPIB or VXI-11, change **Software Connection Name** to an option that corresponds to the control interface with which you are working.



- Click the **Start Demo** button. The **Scope Analysis Demo** window appears and defaults to the **Acquire** tab. The demo locates a signal on Channel 1, as shown in the following example. A trace/signal appears on the waveform graph.



- Use the **Acquire**, **Analyze**, and **Present** tabs to navigate the demo and to configure the signal processing, report generation, and analysis options. Use the knobs and buttons on the scope to configure the horizontal and vertical parameters of the acquisition data. For example, when you change the horizontal scale on the scope, the waveform graph in the demo changes.
- Click the **Analyze** tab. The graph on the upper portion of the window displays a histogram of the signal on Channel 1. Experiment by selecting different analysis options from the list on the lower left corner of the window. The displays and controls change based on your selections.
- Select the **Statistics/Histogram** analysis option.

8. Use the **Intervals** control to set the number of histogram intervals. Change the number of intervals from 128 to 256 and notice the increase in resolution on the x-axis of the graph.
9. Select the **Spectral Analysis** analysis option. If you are examining a repeating, triggered signal, change the **Averaging Mode** control to enable averaging.



Note Refer to the *LabVIEW Measurements Manual*, available online at ni.com in PDF format, for more information about analysis tools.

10. Click the **Present** tab. The graph for the analysis type you selected on the **Analyze** tab expands to fill the screen.
11. Click the **Export HTML Snapshot** button and enter a valid filename with a `.htm` or `.html` extension in the dialog box that appears. The application takes a screenshot of the window, embeds the graphic in the HTML file, and displays the file in the default Web browser.

Virtual Instruments

This chapter guides you through finding examples in LabVIEW. It also teaches you how to create several VIs in LabVIEW.

You will learn to do the following:

- Find and run an example.
- Create a new VI in LabVIEW. The VI you build generates data, analyzes it, and writes it to a file.

Search for Examples

After completing the *LabVIEW Tutorial*, you are ready to run some examples in LabVIEW using the Example Browser, which is part of the LabVIEW Add-On for Tektronix Open Windows Oscilloscopes.

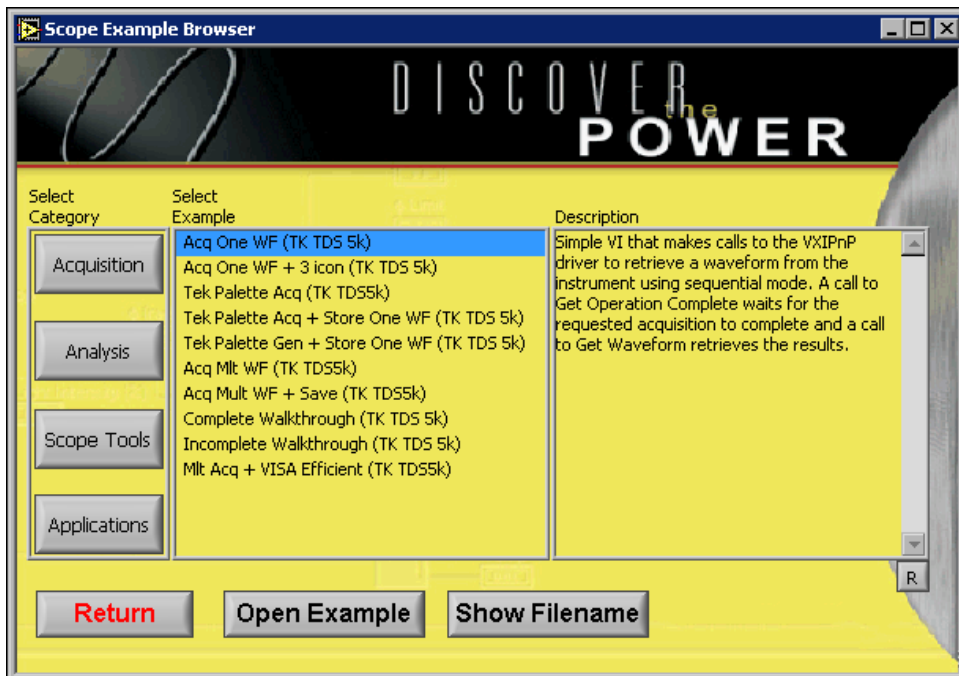
Search for Scope-Specific Examples

This activity shows you how to call the Example Browser to find scope-related examples.



You can complete this activity in approximately 10 minutes.

1. Create a new VI. In the new VI, select **Help»Scope Example Browser**. The Example Browser appears, as shown in the following illustration.



2. Select **Acq Mlt WF (TK TDS 5k)** from the **Select Demo** list. When you select a VI from the list, a description of that VI appears in the **Description** text box.
3. Click the **Open Demo** button to open the Acq Mlt WF (TK TDS 5k) example.



Tip You also can open an example by double-clicking it from the **Select Demo** list.



Note If you are not using a TDS 5000 series oscilloscope, substitute the appropriate series number in the previous example.

You can run the example VI and examine the block diagram to learn how it works. The Example Browser continues to run as you work, allowing you to select other examples easily.

4. Click the **Analysis** button on the left of the Example Browser. To help you find the examples you need, the Example Browser divides the examples into four categories corresponding to the buttons in the **Select Category** section. Clicking any of these buttons displays the example VIs available in that category.

5. Select an example and click the **Show Filename** button. A dialog box appears and displays both the name of the example as it appears in the **Select Demo** list and the full directory path to the example. You can use the full path to load and run examples directly from within the LabVIEW development environment.
6. Click the **Return** button to close the Example Browser.

Finding General LabVIEW Examples

You can use the scope Example Browser to locate and examine scope-specific examples. LabVIEW includes other examples that demonstrate LabVIEW functionality.

To locate general LabVIEW examples, select **Help»Examples** from an open VI. You also can click the **Search Examples** button from the dialog box that appears when LabVIEW opens. To access this dialog box if LabVIEW is running, close all open VIs and the Example Browser.



Note In LabVIEW 6.1, the **Find Examples** dialog box does not fit on the 640 × 480 default screen resolution for a scope. You can use the **Find Examples** dialog box by attaching an external monitor to the scope and enabling it as a secondary monitor. Refer to the Tektronix documentation for information about installing a secondary monitor.

Build a Virtual Instrument

This activity shows how you can build a VI that acquires, plots, and saves oscilloscope data using LabVIEW-based file I/O.



You can complete this activity in approximately 30 minutes.

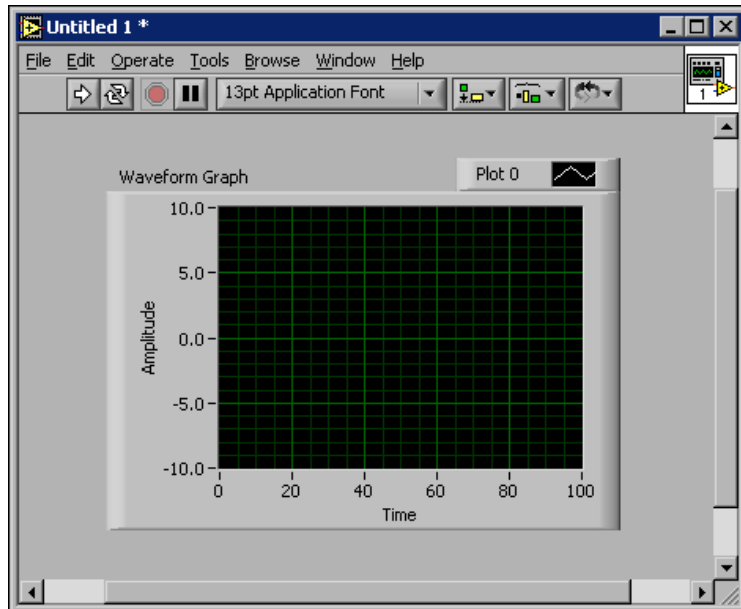
Create a User Interface

Complete the following steps to create a graphical user interface on the front panel using controls and indicators on the **Controls** palette.



You can complete this section in approximately 5 minutes.

1. Create a new VI by clicking the **New VI** button in the **LabVIEW** dialog box.
2. Select a waveform graph on the **Controls»Graph** palette and place it on the front panel, as shown in the following example.



3. Select the Positioning tool, shown at left, on the **Tools** palette and use it to move or resize the waveform graph. Select **Window»Show Tools Palette** to display the **Tools** palette.

Build the Block Diagram

You defined how the user interface will appear. Complete the following steps to create the source code on the block diagram.



You can complete this section in approximately 5 minutes.



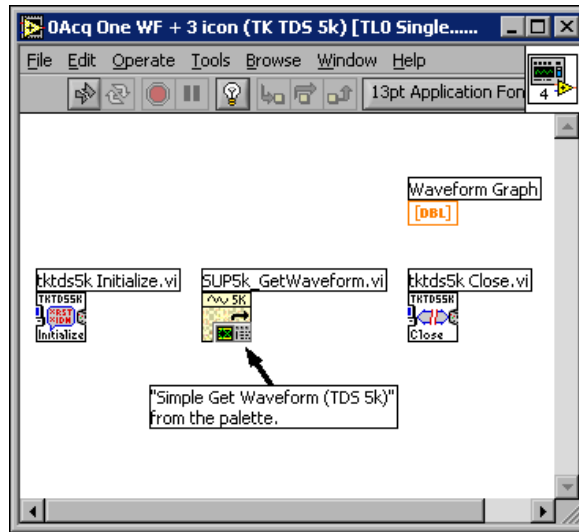
1. Display the block diagram by selecting **Window»Show Diagram** or clicking the block diagram window. The terminal on the block diagram, shown at left, corresponds to the waveform graph on the front panel.



Tip Press the <Ctrl-E> keys to switch between the front panel and block diagram.

2. Select **Window»Show Functions Palette** to display the **Functions** palette.
3. On the **Functions»Tektronix** palette, select the palette icon that matches the scope you are using. For example, if you have a scope in the TDS 5000 series, select the **TDS5k** palette icon.

- Select the Initialize, Simple Get Waveform, and Close VIs on the **Functions»Tektronix»TDS#k** palette and place them on the block diagram, as shown in the following example.



- Select **Window»Show Tools Palette** to display the **Tools** palette.
- Select the Wiring tool, shown at left, on the **Tools** palette.
- Right-click the **reset device** input on the left side of the Initialize VI and select **Create»Constant** from the shortcut menu.
- Use the Operating tool, shown at left, to click the **reset device** constant and set it to FALSE.
- Use the Wiring tool to right-click the **VISA resource name** input on the left side of the Initialize VI and select **Create»Constant** from the shortcut menu.
- Use the Operating tool to set the **VISA resource name** constant to **GPIB8::1::INSTR**. If **GPIB8::1::INSTR** does not appear, make sure the TekScope application is open.



Note The **VISA resource name** is an identifier used by LabVIEW and other applications to specify a software-based connection with the scope. `GPIB8::1::INSTR` is the default identifier for applications running on the scope. The identifier and the software interface that VISA uses to connect with the scope are supplied by the TekScope application. If you close that application, the identifier will not appear as an option.

Wire and Run the VI

You transfer data between block diagram objects through wires. Wires are different colors, styles, and thicknesses, depending on their data types. A broken wire appears as a dashed black line. When you finish wiring this VI, you can run it from the front panel to see it plot data to the chart.



You can complete this section in approximately 5 minutes.

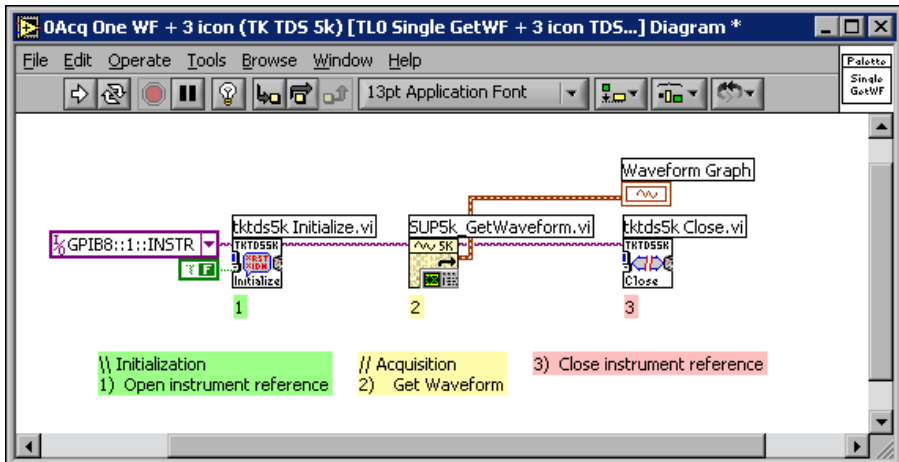


1. Use the Wiring tool, shown at left, to wire the **Waveform** output terminal of the Get Waveform VI to the **Waveform Graph** indicator terminal. To do so, click the **Waveform** output terminal, move the cursor to the **Waveform Graph** indicator terminal to create a temporary wire, and click the indicator terminal to connect the wire.



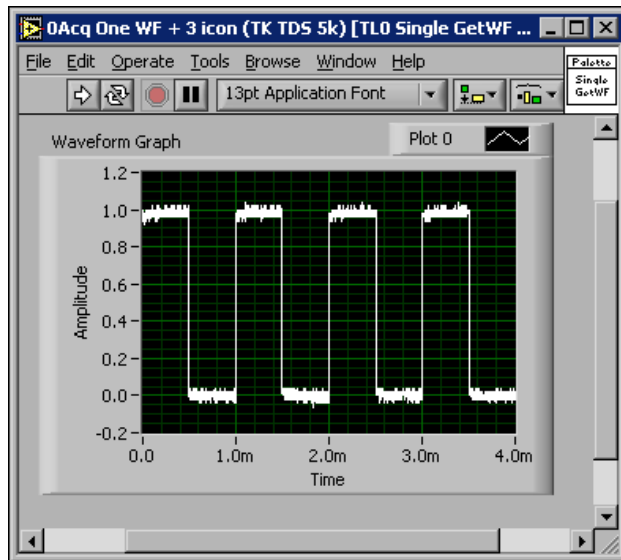
Tip When you move the Wiring tool over a terminal, the terminal blinks and a tip strip with the name of the terminal appears.

2. Wire the **Instrument Handle Out** terminal of the Initialize VI to the **Instrument Handle In** terminal of the Get Waveform VI.
3. Wire the **Instrument Handle Out** terminal of the Get Waveform VI to the **Instrument Handle** terminal of the Close VI. The block diagram should appear similar to the following example.



4. Display the front panel by pressing the <Ctrl-E> keys.

5. Set the scope to acquire a repeated, triggered signal on channel 1. To do so, you can connect scope input to the **Probe Comp** output on the scope and click the **Autoset** button on the main scope user interface.
6. Click the **Run** button, shown at left, on the toolbar to run the VI. If the **Run** button appears broken, click it to see a list of errors that prevent the VI from running.
7. The VI stops running automatically. The front panel should appear similar to the following example.



Note The VI you created in this activity should be similar to the Acq One WF + 3icon (TK TDS#k) example VI located in the Example Browser.

Acquire and Display Waveform Data

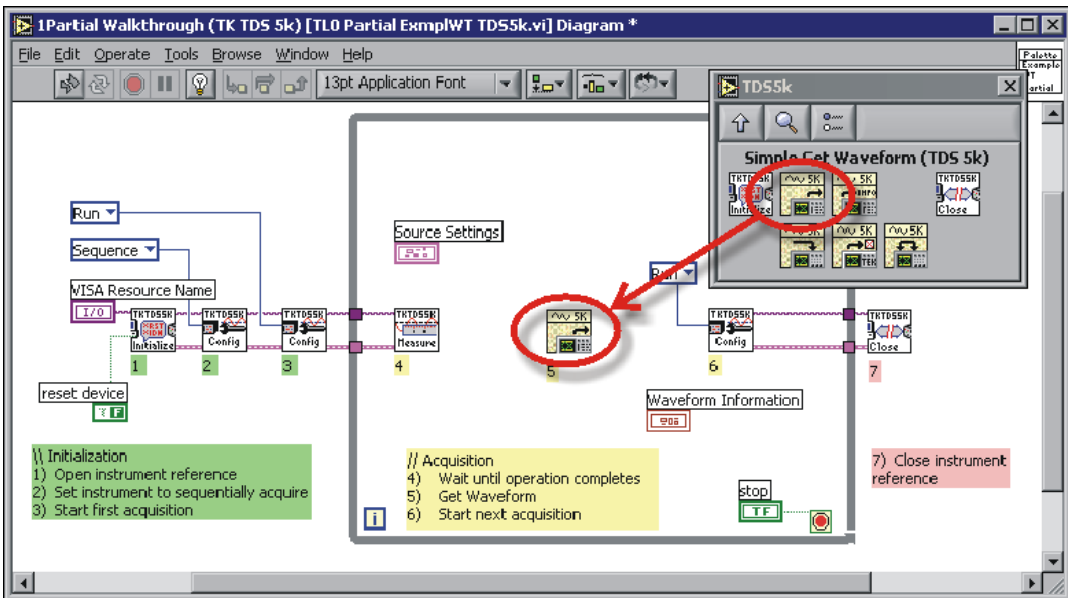
Complete the following steps to create a VI that allows you to control the way waveform data from a scope displays on a graph.



You can complete this section in approximately 10 minutes.

1. Close all open VIs to display the **LabVIEW** dialog box. Click the **New VI** button.
2. Select **Help»Scope Example Browser** to open the Example Browser.
3. Open the Partial Walkthrough (TDS #k) VI.

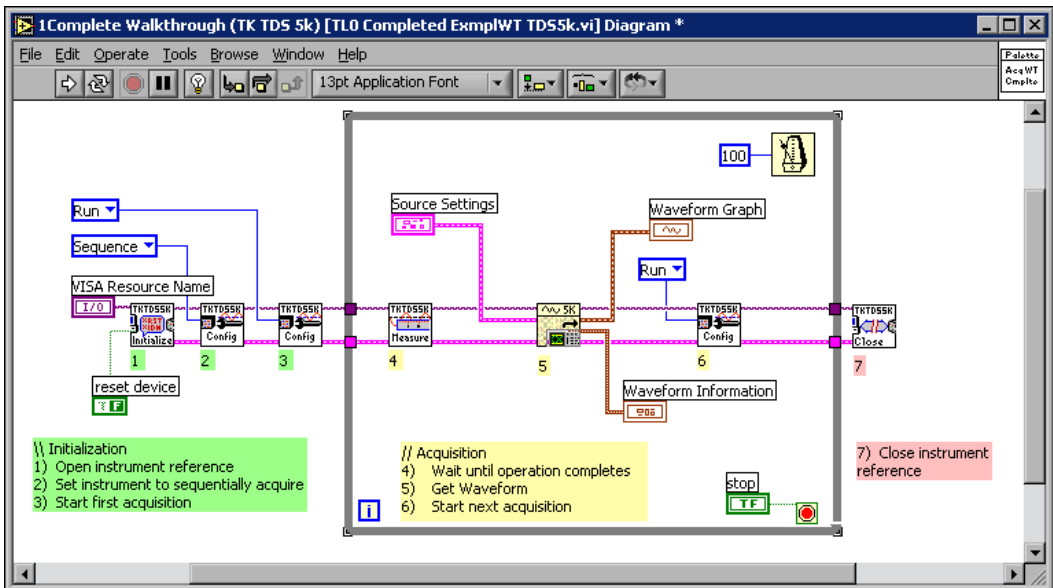
4. Select **File»Save As** to save a copy of the VI as Walkthrough Exercise.vi.
5. Place a waveform graph indicator on the front panel.
6. Press the <Ctrl-E> keys to display the block diagram.
7. Select **Window»Show Functions Palette** to display the **Functions** palette.
8. Select the Get Waveform VI on the **Functions»Tektronix»TDS#k** palette and place it on the block diagram. The block diagram should appear similar to the following example.



9. Use the Wiring tool to wire the **Instrument Handle Out** terminal of the Get Operation Complete VI to the **Instrument Handle In** terminal of the Get Waveform VI. Wire the **Instrument Handle Out** terminal of the Get Waveform VI to the **Instrument Handle In** terminal of the Set Acquisition VI. The wires pass a valid instrument handle to each subVI. The instrument handle is a unique identifier that LabVIEW uses to maintain a software connection with the oscilloscope.
10. Wire the **error out** terminal of the Get Operation Complete VI to the **error in** terminal of the Get Waveform VI. Wire the **error out** terminal of the Get Waveform VI to the **error in** terminal of the Set Acquisition VI. The wires ensure that LabVIEW can handle errors that occur when the VI runs. For example, if the Initialize VI receives an invalid

instrument description string, its **error out** output indicates why the VI did not attempt to retrieve waveform data.

11. Wire the **Source Settings** control terminal to the **Source Settings** input of the Get Waveform VI. The control terminal specifies the source of the waveform data. The control is interactive so the user can change the source while the VI runs.
12. Wire the **Waveform** output of the Get Waveform VI to the **Waveform Graph** indicator terminal.
13. Wire the **Waveform Information** output of the Get Waveform VI to the **Waveform Information** indicator terminal. The wires pass the waveform, scaling, and offset information to front panel indicators that display the information graphically. The block diagram should appear similar to the following example.



14. Display the front panel and click the **Run** button, shown at left, to run the VI.
15. Click the **Stop** button on the front panel to stop the VI.
16. Save the VI.



Note The VI you created in this activity should be similar to the Complete Walkthrough (TK TDS #k) example VI located in the Example Browser.

Scope-Specific Programming

This chapter describes programming techniques to help you create custom VIs for oscilloscopes.

Using Scope-Specific VIs

Options for LabVIEW Communication with a Scope

When you create a LabVIEW application, you can select any of the following ways in which VIs and subVIs communicate with the scope hardware:

- Invoke calls to a LabVIEW-based instrument driver.
- Send and receive text-based messages through standard Virtual Instrument Software Architecture (VISA) calls. Use this method for a high degree of control over scope functionality. This method has a steep learning curve, but can reduce your test time by optimizing your applications.
- Invoke a combination of calls to either or both of the two previous options.
- Invoke a combination of calls to VIs on the **Functions»Tektronix** palette, which installs with the LabVIEW Add-On for Tektronix Open Windows Oscilloscopes, and to either or both of the first two options. Use this method to acquire and/or store waveforms with a minimum of coding.

Programming with Scope-Specific VIs

You can use the VIs located on the **Functions»Tektronix** palette to perform some of the tasks associated with transferring and manipulating waveforms between your LabVIEW application and the scope. The VIs communicate with the scope through the standard Tektronix *VXIplug&play* Instrument Driver and simplify the following tasks:

- Transfer a waveform and its timing/scaling information to a LabVIEW waveform and cluster.
- Transfer waveform timing/scaling information to a LabVIEW cluster.

- Transfer a waveform and its timing/scaling information from a LabVIEW waveform data type to a scope reference slot or a file.
- Transfer a waveform from one location on the scope, such as a channel, reference slot, or file, to another location on the scope, such as a reference slot or file.

The scope-specific VIs simplify the previous tasks by grouping related operations. For example, the Get Waveform VI can acquire waveform signals either from a live source, such as a channel, reference, or math source, or from a file. A single call to the Store Waveform VI can populate either a scope-based waveform reference or a file.

In contrast, the *VXIplug&play* driver contains individual functions for each source/target combination. By using the scope-specific VIs, you can reduce the number of steps necessary to implement the functionality you need when programming the scope.

Measurement-related data types, such as the waveform data type and pre-defined clusters, also can simplify tasks. These data types combine the numeric values of a waveform, the sampling rate, x/y offsets, the x/y scaling factors, and other signal-related information. VIs that directly invoke the *VXIplug&play* driver generally require more wiring because you track information manually using separate wires.

Scope-specific VIs also can function as a teaching tool. To examine the block diagram of a VI, right-click the VI on the **Functions** palette and select **Open VI** from the shortcut menu. On the front panel, select **Window»Show Diagram** to display the block diagram.

Commonly Used Scope-Specific VIs

The VIs on the **Functions»Tektronix** palette share some common elements that you might want to use in development. This section reviews some of these core elements.

Initialize and Close

Initialize and Close VIs manipulate a software connection with the scope. You can use the VIs before and after other VIs that communicate with the scope. The Initialize VI establishes a connection using a resource name or an instrument descriptor, which uniquely describes a particular instrument. The instrument descriptor that identifies the scope hardware for code that runs directly on the scope is `GPIB8 : : 1 : : INSTR`. If you work with NI-VISA, you can use Measurement & Automation Explorer to determine

and configure other instrument descriptors. Select **Tools»Measurement & Automation Explorer**.

Get and Set Waveform

Although LabVIEW Scope Add-On VIs automatically acquire and store waveforms between a VI and a scope, you might consider manipulating the waveforms manually if you work directly with the functions of the *VXIplug&play* driver.

For example, the Get Waveform VI included with the *VXIplug&play* driver transfers samples from the scope to LabVIEW. You must specify the number of samples to expect in the acquisition to facilitate memory allocation. Use the Get Horizontal Reclength or Get Horizontal Parameters VIs included with the *VXIplug&play* driver to determine the appropriate value to use before acquiring waveform data.

Set the display status of a channel to **on** before the Get Waveform function retrieves data. The Get Waveform function cannot run until references are displayed and occupied. Use the *VXIplug&play* function Set Wfm Display Status to change the display status of a channel or reference.

Additional Scope-Specific Programming Information

The following information sources are related to scope-specific programming:

- Context-sensitive help associated with the VIs located on the LabVIEW Scope Add-On Palette and the *VXIplug&play* Instrument Driver—Select **Help»Context Help** or press the <Ctrl-H> keys to display the **Context Help** window. The VI description appears in the **Context Help** window when you move the cursor over the VI icon.
- Text files located on the LabVIEW Scope Add-On CD—Open the `readme1st.htm` file on the CD to display a document that lists the available text files.

Debugging

This chapter teaches you how to use some of the debugging techniques available in LabVIEW.

Use Execution Highlighting

Execution highlighting shows the movement of data on the block diagram from one node to another using bubbles that move along the wires.



You can complete this activity in approximately 5 minutes.



1. Open the Walkthrough Exercise VI you created in the *Acquire and Display Waveform Data* section of Chapter 3, *Virtual Instruments*.
2. On the oscilloscope, use a probe to connect Channel 1 to the **Probe Compensation** signal.
3. Display the block diagram and click the **Highlight Execution** button, shown at left, on the toolbar.
4. Click the **Run** button, shown at left, to run the VI from the block diagram. The VI runs in slow motion with moving bubbles to highlight the flow of execution. It also displays data as it becomes available in the VI.
5. Click the **Stop** button on the front panel to stop the VI.
6. Click the **Highlight Execution** button again to disable execution highlighting.

Single-Step with Probes

You can set breakpoints in a VI and single-step into, out of, and over sections of the block diagram. You also can insert probes to examine data values in detail during program execution.

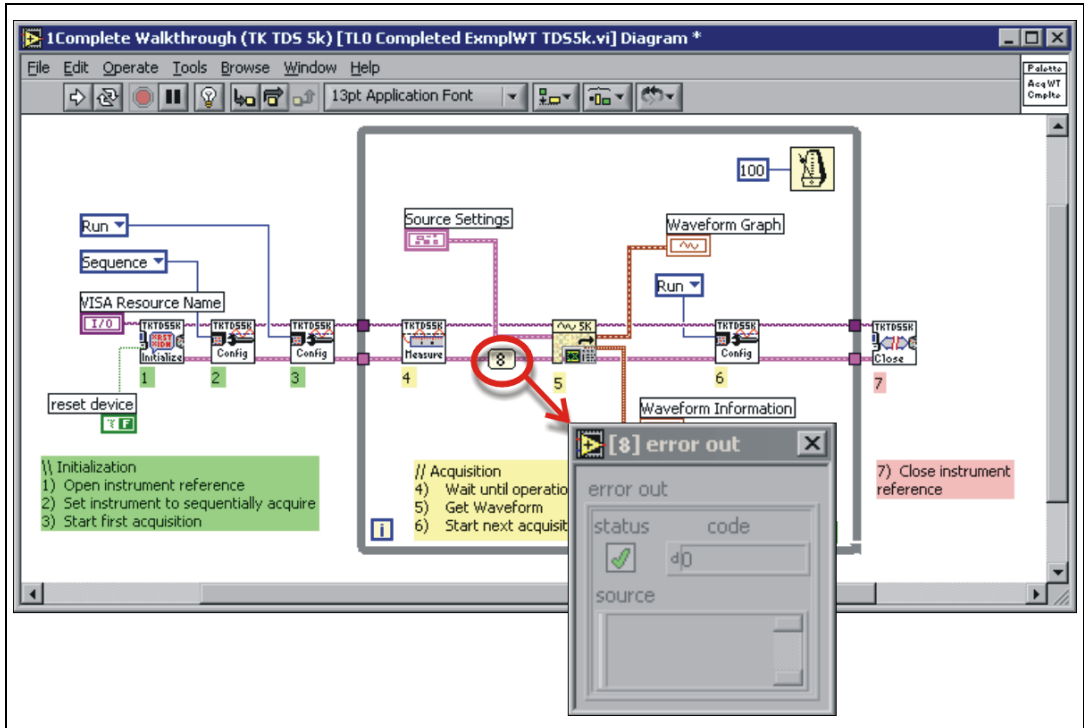
Complete the following steps to insert probes and single-step through a VI to monitor data while the VI runs.



You can complete this activity in approximately 10 minutes.

1. Open the Walkthrough Exercise VI you created in the *Acquire and Display Waveform Data* section of Chapter 3, *Virtual Instruments*.
2. Make sure that the **VISA resource name** control is set to the name that designates the scope you are using. The default value of GPIB8 : : 1 : : INSTR works when you are working directly on the scope.
3. On the scope, use a probe to connect Channel 1 to the Probe Compensation signal.
4. Display the block diagram.
5. Select the Breakpoint tool, shown at left, from the **Tools** palette.
6. Click the block diagram on an open area in the While Loop. A red border appears in the While Loop to indicate that you set a breakpoint for the loop.
7. Click the **Run** button. The VI runs up to the breakpoint you set, then pauses. The red border on the While Loop flashes to indicate the location of the pause and that the VI is running in single-step mode.
8. Click the **Step Into** button, shown at left, on the toolbar. The Get Operation Complete VI flashes to indicate that execution paused in the While Loop and before the Get Operation Complete VI runs.
9. Right-click the wire out of the **error out** terminal on the Get Operation Complete VI and select **Probe** from the shortcut menu. A window displays the values that the **error out** terminal contains, as shown in the following example. In single-step mode, you can use the Probe tool to examine data from the wires of the block diagram as the data becomes available. The **probe** window contains no data because the Get Operation Complete VI has not generated information to populate it.





10. Click the **Step Over** button, shown at left, to execute the Get Operation Complete VI. The **error out** information now appears in the probe window.



11. Click the **Pause** button, shown at left, to run the VI in normal mode until the next breakpoint. The VI runs until it reaches the breakpoint you set in step 6.

12. Use the Breakpoint tool to click an open area in the While Loop to clear the breakpoint you set in step 6.

13. Click the **Pause** button to run the VI in normal mode.

14. Display the front panel.

15. Click the **Stop** button.

Where to Go from Here

This chapter describes how to use the *LabVIEW Help* and the additional resources available to you.

Refer to the *LabVIEW User Manual*, the *LabVIEW Measurements Manual*, or the *LabVIEW Help* for more information about LabVIEW programming and the measurement-oriented tools that are part of LabVIEW.

You can use the *LabVIEW Bookshelf* to search Portable Document Format (PDF) versions of all the LabVIEW manuals and Application Notes. Access the *LabVIEW Bookshelf* by selecting **Help»View Printed Manuals**. You must have Adobe Acrobat Reader 3.0 or later installed to view these files.

Online Help

The *LabVIEW Help* contains descriptions of all built-in LabVIEW VIs and functions and their parameters. Access the *LabVIEW Help* by selecting **Help»Contents and Index**. You also can right-click any built-in VI or function on the block diagram and select **Help** from the shortcut menu. Access context-sensitive help by selecting **Help»Show Context Help** or pressing the <Ctrl-H> keys.



When you place a VI or function on the block diagram and move the cursor over it, a description of the object appears in the **Context Help** window.



Click the lock button at the bottom of the **Context Help** window to lock the current contents of the window.

Web Resources

The NI Developer Zone at zone.ni.com is the essential resource for building measurement and automation systems. At the NI Developer Zone, you can easily access the latest example programs, system configurators, tutorials, technical news, as well as a community of developers ready to share their own techniques.

The LabVIEW/Tektronix resources page at ni.com/tek provides support resources for using LabVIEW with Tektronix Open Windows oscilloscopes. The page also provides information on purchasing the LabVIEW Full and Professional Development Systems.

National Instruments' Commitment to You

LabVIEW represents a long-standing commitment by National Instruments to provide tools that simplify the development of instrumentation, data acquisition, and control systems. When you choose LabVIEW as your development environment, you join thousands of scientists and engineers who are taking advantage of the power of graphical programming.

Customer Education

For additional training, National Instruments offers interactive CDs, videos, books, and hands-on LabVIEW courses to help you master LabVIEW quickly and develop successful applications.

Alliance Program

The Alliance Program is a network of third-party developers and consultants who are experts in LabVIEW and other National Instruments products. The National Instruments *Alliance Solutions* directory lists additional libraries and utilities developed by our Alliance members to help you use LabVIEW. In addition, the *Alliance Solutions* directory lists expert LabVIEW consultants who can help you develop custom applications.

System Requirements

Driver Requirements

The factory-installed software on the scope includes several drivers that are necessary for communication between LabVIEW and your scope hardware. Review the software driver requirements if you want to install a non-evaluation version of LabVIEW that you purchased, need more details about how the software works, or otherwise want to reconfigure software on the scope.

Most applications require an instrument driver, a VISA driver, and a GPIB driver. Refer to the `readme.htm` file on the installation CD for minimum version recommendations and complete installation instructions.

Other Operating System Requirements

You can run LabVIEW on Windows, Mac OS, Linux, and UNIX platforms. The following table describes the required system configurations for running LabVIEW on these platforms.

Platform	Media and System Requirements	Important Notes
All Platforms	Distributed on CD-ROM.	LabVIEW and the <i>LabVIEW Help</i> contain 16-bit color graphics. LabVIEW requires a minimum color palette setting of 256 colors. 16-bit color is recommended. The <i>LabVIEW Help</i> requires a minimum color palette setting of 256 colors with a screen resolution of 1,024 × 768 pixels. 16-bit color is recommended.

Platform	Media and System Requirements	Important Notes
All Windows Versions	<p>Refer to the installation instructions that appear on your screen for information on the size of the LabVIEW system you are installing.</p> <p>LabVIEW runs on any system that supports Windows.</p>	<p>To use Measurement & Automation Explorer, you must have Microsoft Internet Explorer 5.0 or later installed.</p> <p>The <i>LabVIEW Tutorial</i> requires a sound card, a video card capable of playing .avi files, and a minimum color palette setting of 256 colors with a screen resolution of 1024 × 768 pixels. 16-bit color is recommended.</p>
Windows NT	LabVIEW runs on Windows NT 4.0 Service Pack 3 or later.	To take advantage of ActiveX functionality in LabVIEW 6.0, you must have Windows NT 4.0 Service Pack 3 or later and Microsoft Internet Explorer 4.0 or later.
Power Macintosh	<p>LabVIEW requires System 7.6.1 or later.</p> <p>You need a minimum of 32 MB of RAM and at least 100 MB of disk storage space for the minimal installation of LabVIEW or 250 MB for the full installation.</p>	<p>National Instruments recommends that you have at least 32 MB of RAM. You might need more memory, depending on the size of the application you design in LabVIEW and the amount of data that your application manipulates.</p> <p>For more accurate timing, install the Apple QuickTime extension. When you use QuickTime, timing accuracy should increase from 16.6 ms resolution to approximately 1 ms resolution. System response varies depending on background applications, other extensions, networking activity, and disk caching.</p> <p>The <i>LabVIEW Tutorial</i> requires a sound card, a video card capable of playing .avi files, and a minimum color palette setting of 256 colors with a screen resolution of 1024 × 768 pixels. 16-bit color is recommended.</p>

Platform	Media and System Requirements	Important Notes
All UNIX Versions	<p>LabVIEW requires an X Window System server, such as OpenWindows, CDE, or X11R6.</p> <p>You need a minimum of 32 MB of RAM with 32 MB of swap space storage.</p> <p>You need between 65 MB to 150 MB of disk storage space depending on the components you install.</p>	<p>LabVIEW uses a directory for storing temporary files. Some of the temporary files are large, so keep several megabytes of disk space available for this temporary directory. The default for the temporary directory is <code>/tmp</code>. You can change the temporary directory by selecting Tools»Options.</p> <p>If LabVIEW aborts unexpectedly, it might leave files behind in the temporary directory. Remove old files occasionally to avoid depleting your disk space.</p> <p>To save space, install only the VIs you plan to use.</p> <p>LabVIEW does not require a specific graphical user interface (GUI) such as Motif or OpenLook, because LabVIEW uses <code>xlib</code> to create its own GUI.</p>
Sun	LabVIEW runs on SPARCstations with Solaris 2.5.1 or later.	—
Linux	LabVIEW runs on Linux for Intel x86 processors with kernel version 2.0.x or later. LabVIEW runs on most major Linux distributions, such as RedHat, Caldera, SuSE, and Debian.	<p>Requires GNU C Library Version 2 (<code>glibc2</code>, also known as <code>libc.so.6</code>).</p> <p>RedHat Linux 5.0 or later includes the <code>glibc2</code> run-time library.</p>

Technical Support and Professional Services

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

- **Support**—Online technical support resources include the following:
 - **Self-Help Resources**—For immediate answers and solutions, visit our extensive library of technical support resources available in English, Japanese, and Spanish at ni.com/support. These resources are available for most products at no cost to registered users and include software drivers and updates, a KnowledgeBase, product manuals, step-by-step troubleshooting wizards, hardware schematics and conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, a measurement glossary, and so on.
 - **Assisted Support Options**—Contact NI engineers and other measurement and automation professionals by visiting ni.com/ask. Our online system helps you define your question and connects you to the experts by phone, discussion forum, or email.
- **Training**—Visit ni.com/custed for self-paced tutorials, videos, and interactive CDs. 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, NI Alliance Program members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Glossary

B

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

breakpoint Pause in execution used for debugging.

Breakpoint tool Tool to set a breakpoint on a VI, node, or wire.

C

control Front panel object for entering data to a VI interactively or to a subVI programmatically, such as a knob, push button, or dial.

Controls palette Palette that contains front panel controls, indicators, and decorative objects.

D

data flow Programming system that consists of executable nodes that execute only when they receive all required input data and produce output automatically when they execute. LabVIEW is a dataflow system.

E

execution highlighting Debugging technique that animates VI execution to illustrate the data flow in the VI.

F

front panel Interactive user interface of a VI. Front panel appearance imitates physical instruments, such as oscilloscopes and multimeters.

function Built-in execution element, comparable to an operator, function, or statement in a text-based programming language.

Functions palette Palette that contains VIs, functions, block diagram structures, and constants.

G

General Purpose Interface Bus GPIB—synonymous with HP-IB. The standard bus used for controlling electronic instruments with a computer. Also called IEEE 488 bus because it is defined by ANSI/IEEE Standards 488-1978, 488.1-1987, and 488.2-1992.

I

IEEE 488.2 Institute of Electrical and Electronic Engineers Standard 488.2-1987, which defines the GPIB.

indicator Front panel object that displays output, such as a graph or LED.

L

Labeling tool Tool to create labels and enter text into text windows.

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

N

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.

O

Operating tool Tool to enter data into controls and operate them.

P

palette	Display of icons that represent possible options.
Positioning tool	Tool to move and resize objects.
probe	Debugging feature for checking intermediate values in a VI.
PXI	PCI eXtensions for Instrumentation. A modular, computer-based instrumentation platform.

R

RS-232	Recommended Standard 232, a serial interface bus standard.
RS-485	Recommended Standard 485, a serial interface bus standard.

S

structure	Program control element, such as a Sequence structure, Case structure, For Loop, or While Loop.
subpalette	A palette contained in an icon of another palette.
subVI	VI used in the block diagram of another VI. Comparable to a subroutine.

T

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.
Tools palette	Palette that contains tools you can use to edit and debug front panel and block diagram objects.

V

VI	<i>See</i> virtual instrument.
virtual instrument	Program in LabVIEW that models the appearance and function of a physical instrument.
VISA	Single interface library for controlling GPIB, VXI, RS-232, and other types of instruments.

W

wire	Data path between nodes.
wire branch	Section of wire that contains all the wire segments from junction to junction, terminal to junction, or terminal to terminal if there are no junctions between.
wire junction	Point where three or more wire segments join.
wire segment	Single horizontal or vertical piece of wire.
Wiring tool	Tool to define data paths between terminals.