

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

LabVIEW™

LabVIEW Fundamentals

August 2005
374029A-01

NATIONAL INSTRUMENTS™

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Worldwide Technical Support and Product Information

ni com

National Instruments Corporate Headquarters

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

Worldwide Offices

Australia 1800 300 800, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 6555 7838, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, India 91 80 51190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400, Lebanon 961 0 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 0 348 433 466, New Zealand 0800 553 322, Norway 47 0 66 90 76 60, Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 095 783 68 51, Singapore 1800 226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 02 2377 2222, Thailand 662 278 6777, United Kingdom 44 0 1635 523545

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

© 2005 National Instruments Corporation. All rights reserved.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Important Information

Warranty

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

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

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

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

Copyright

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

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

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

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

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright © 1998, 1999, 2000, 2001, 2003 by the Board of Trustees of the University of Illinois. All rights reserved.

Trademarks

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

FireWire® is the registered trademark of Apple Computer, Inc. Other product and company names mentioned herein are trademarks or trade names of their respective companies.

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

Patents

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Contents

About This Manual

Conventions xiii

Chapter 1
Introduction to LabVIEW

LabVIEW Documentation Resources 1-1
LabVIEW Help 1-1
Print Documents 1-2
Readme Documents 1-3
LabVIEW VI Templates, Example VIs, and Tools 1-3
LabVIEW VI Templates 1-4
LabVIEW Example VIs 1-4
LabVIEW Tools for DAQ Configuration (Windows) 1-4

Chapter 2
Introduction to Virtual Instruments

Front Panel 2-2
Block Diagram 2-2
Terminals 2-3
Nodes 2-4
Wires 2-4
Structures 2-5
Icon and Connector Pane 2-5
Using and Customizing VIs and SubVIs 2-6

Chapter 3
LabVIEW Environment

Getting Started Window 3-1
Controls Palette 3-1
Functions Palette 3-2
Navigating the Controls and Functions Palettes 3-2
Tools Palette 3-3
Menus and Toolbars 3-4
Menus 3-4
Shortcut Menus 3-4
VI Toolbar 3-5
Project Explorer Window Toolbars 3-5

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Context Help Window 3-5
Project Explorer Window 3-6
Navigation Window 3-6
Customizing Your Work Environment 3-7
Customizing the Controls and Functions Palettes 3-7
Setting Work Environment Options 3-7

Chapter 4
Building the Front Panel

Front Panel Controls and Indicators 4-1
Styles of Controls and Indicators 4-2
Modern and Classic Controls and Indicators 4-2
System Controls and Indicators 4-2
Numeric Displays, Slides, Scroll Bars, Knobs, Dials, and Time Stamps 4-2
Numeric Controls and Indicators 4-3
Slide Controls and Indicators 4-3
Scroll Bar Controls and Indicators 4-4
Rotary Controls and Indicators 4-4
Time Stamp Control and Indicator 4-4
Graphs and Charts 4-5
Buttons, Switches, and Lights 4-5
Radio Buttons Controls 4-5
Text Entry Boxes, Labels, and Path Displays 4-6
String Controls and Indicators 4-6
Combo Box Controls 4-6
Path Controls and Indicators 4-7
Array, Matrix, and Cluster Controls and Indicators 4-7
Listboxes, Tree Controls, and Tables 4-7
Listboxes 4-7
Tree Controls 4-7
Tables 4-8
Ring and Enumerated Type Controls and Indicators 4-8
Ring Controls 4-8
Enumerated Type Controls 4-8
Container Controls 4-9
Tab Controls 4-9
Subpanel Controls 4-9
I/O Name Controls and Indicators 4-10
Waveform Control 4-10
Digital Waveform Control 4-10
Digital Data Control 4-11
References to Objects or Applications 4-11
.NET and ActiveX Controls (Windows) 4-12

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Configuring Front Panel Objects 4-12
Showing and Hiding Optional Elements 4-13
Changing Controls to Indicators and Indicators to Controls 4-13
Replacing Front Panel Objects 4-13
Configuring the Front Panel 4-13
Coloring Objects 4-14
Aligning and Distributing Objects 4-14
Grouping and Locking Objects 4-14
Resizing Objects 4-15
Adding Space to the Front Panel without Resizing the Window 4-15
Labeling 4-16
Text Characteristics 4-16
Designing User Interfaces 4-17
Using Front Panel Controls and Indicators 4-17
Designing Dialog Boxes 4-17

Chapter 5
Building the Block Diagram

Block Diagram Objects 5-1
Block Diagram Terminals 5-1
Control and Indicator Data Types 5-2
Constants 5-3
Block Diagram Nodes 5-3
Polymorphic VIs and Functions 5-4
Functions Overview 5-4
Adding Terminals to Functions 5-5
Built-in VIs and Functions 5-5
Express VIs 5-5
Using Wires to Link Block Diagram Objects 5-6
Wire Appearance and Structure 5-6
Wiring Objects 5-7
Bending Wires 5-7
Undoing Wires 5-8
Automatically Wiring Objects 5-8
Selecting Wires 5-8
Correcting Broken Wires 5-8
Coercion Dots 5-9
Block Diagram Data Flow 5-9
Data Dependency and Artificial Data Dependency 5-10
Missing Data Dependencies 5-11
Flow-Through Parameters 5-12
Data Flow and Managing Memory 5-12
Designing the Block Diagram 5-13

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 6
Running and Debugging VIs

Running VIs 6-1
Correcting Broken VIs 6-2
Finding Causes for Broken VIs 6-2
Common Causes of Broken VIs 6-3
Debugging Techniques 6-3
Execution Highlighting 6-3
Single-Stepping 6-4
Probe Tool 6-4
Breakpoints 6-4
Handling Errors 6-5
Error Clusters 6-6
Using While Loops for Error Handling 6-7
Using Case Structures for Error Handling 6-7

Chapter 7
Creating VIs and SubVIs

Searching for Examples 7-1
Using Built-In VIs and Functions 7-1
Creating SubVIs 7-1
Creating an Icon 7-2
Building the Connector Pane 7-3
Creating SubVIs from Sections of a VI 7-4
Designing SubVI Front Panels 7-4
Viewing the Hierarchy of VIs 7-4
Polymorphic VIs 7-5
Saving VIs 7-6
Naming VIs 7-6
Saving for a Previous Version 7-7
Customizing VIs 7-7

Chapter 8
Loops and Structures

For Loop and While Loop Structures 8-1
For Loops 8-2
While Loops 8-3
Controlling Timing 8-5
Auto-Indexing Loops 8-5
Auto-Indexing to Set the For Loop Count 8-6
Auto-Indexing with While Loops 8-6

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Loops to Build Arrays....8-7
Shift Registers and the Feedback Node in Loops....8-7
Shift Registers....8-7
Feedback Node....8-10
Default Data in Loops....8-11
Case, Sequence, and Event Structures....8-11
Case Structures....8-11
Case Selector Values and Data Types....8-12
Input and Output Tunnels....8-13
Using Case Structures for Error Handling....8-13
Sequence Structures....8-13
Event Structures....8-15

Chapter 9
Grouping Data Using Strings, Arrays, and Clusters

Grouping Data with Strings....9-1
Strings on the Front Panel....9-1
String Display Types....9-2
Tables....9-2
Editing, Formatting, and Parsing Strings....9-3
Formatting and Parsing Strings....9-3
Grouping Data with Arrays and Clusters....9-4
Arrays....9-4
Restrictions....9-4
Indexes....9-4
Examples of Arrays....9-5
Creating Array Controls, Indicators, and Constants....9-7
Creating Multidimensional Arrays....9-7
Array Functions....9-8
Default Data in Arrays....9-10
Clusters....9-10
Order of Cluster Elements....9-10
Cluster Functions....9-11
Creating Cluster Controls, Indicators, and Constants....9-11

Chapter 10
Graphs and Charts

Types of Graphs and Charts....10-1
Waveform Graphs and Charts....10-2
Waveform Graphs....10-2
Waveform Charts....10-3
Waveform Data Type....10-3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

XY Graphs 10-3
Intensity Graphs and Charts 10-4
Intensity Charts 10-5
Intensity Graphs 10-6
Digital Waveform Graphs 10-7
Digital Waveform Data Type 10-10
3D Graphs 10-10
Customizing Graphs and Charts 10-13
Using Multiple X- and Y-Scales 10-13
Autoscaling 10-13
Formatting X- and Y-Scales 10-13
Using the Graph Palette 10-14
Customizing Graph and Chart Appearance 10-15
Customizing Graphs 10-15
Using Graph Cursors 10-16
Using Graph Annotations 10-17
Customizing 3D Graphs 10-18
Customizing Charts 10-18
Configuring Chart History Length 10-19
Configuring Chart Update Modes 10-19
Using Overlaid and Stacked Plots 10-20

Chapter 11
File I/O

Basics of File I/O 11-1
Choosing a File I/O Format 11-2
Using VIs and Functions for Common File I/O Operations 11-3
Using Storage VIs 11-5
Creating Text and Spreadsheet Files 11-6
Formatting and Writing Data to Files 11-7
Scanning Data from Files 11-7
Creating Binary Files 11-7
Creating Datalog Files 11-7
Writing Waveforms to Files 11-8
Reading Waveforms from Files 11-9

Chapter 12
Documenting and Printing VIs

Documenting VIs 12-1
Printing VIs 12-2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix A
Technical Support and Professional Services

Glossary

Index

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

About This Manual

Before you read this manual, use the *Getting Started with LabVIEW* 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 describes LabVIEW programming concepts, techniques, features, VIs, and functions you can use to create test and measurement, data acquisition, instrument control, datalogging, measurement analysis, and report generation applications.

This manual is a subset of the content available in the *LabVIEW Help*, which includes all the content in this manual. Refer to the *LabVIEW Help* for more information about any of the concepts described in this manual.

This manual does not include specific information about each palette, tool, menu, dialog box, control or indicator, or built-in VI or function. Refer to the *LabVIEW Help* for more information about these items and for detailed, step-by-step instructions for using LabVIEW features and for building specific applications. Refer to the *LabVIEW Documentation Resources* section of Chapter 1, *Introduction to LabVIEW*, for more information about the *LabVIEW Help* and accessing it.

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.



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

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

	names, controls and indicators on the front panel, dialog boxes, sections of dialog boxes, menu names, and palette names.
italic	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.
`monospace`	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, operations, variables, filenames, and extensions.
`monospace bold`	Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.
`monospace italic`	Italic text in this font denotes text that is a placeholder for a word or value that you must supply.
Platform	Text in this font denotes a specific platform and indicates that the text following it applies only to that platform.
right-click	**(Mac OS)** Press <Command>-click to perform the same action as a right-click.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1 Introduction to LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine the order of program execution, LabVIEW uses dataflow programming, where the flow of data through the nodes on the block diagram determines the execution order of the VIs and functions. VIs, or virtual instruments, are LabVIEW programs that imitate physical instruments.

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. This graphical source code is also known as G code or block diagram code. The block diagram contains this code. In some ways, the block diagram resembles a flowchart.

You can purchase several add-on software toolkits for developing specialized applications. All the toolkits integrate seamlessly in LabVIEW. Refer to the National Instruments Web site at `ni com\toolkits` for more information about these toolkits.

LabVIEW Documentation Resources

LabVIEW includes extensive online and print documentation for new and experienced LabVIEW users.

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, and tools.

The *LabVIEW Help* includes links to the technical support resources on the National Instruments Web site, such as NI Developer Zone, the KnowledgeBase, and the Product Manuals Library.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Access the *LabVIEW Help* by selecting Help»Search the LabVIEW Help. You also can print a help topic or a book of help topics from the *LabVIEW Help*.

Refer to the *LabVIEW Help* for more information about printing help topics.

Note **(Mac OS)** National Instruments recommends that you use Safari 1.0 or later or Firefox 1.0.2 or later to view the *LabVIEW Help*. **(Linux)** National Instruments recommends that you use Netscape 6.0 or later, Mozilla 1.2 or later, or Firefox 1.0.2 or later to view the *LabVIEW Help*.

After you install a LabVIEW add-on such as a toolkit, module, or driver, the documentation for that add-on appears in the *LabVIEW Help* or appears in a separate help system you access by selecting Help»Add-On Help, where Add-On Help is the name of the separate help system for the add-on.

Print Documents

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

- *Getting Started with LabVIEW*—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.
- *LabVIEW Quick Reference Card*—Use this card as a reference for information about documentation resources, keyboard shortcuts, data type terminals, and tools for editing, execution, and debugging.
- *LabVIEW Fundamentals*—Use this manual to learn about LabVIEW programming concepts, techniques, features, VIs, and functions you can use to create test and measurement, data acquisition, instrument control, datalogging, measurement analysis, and report generation applications. The *LabVIEW Help* includes all the content in 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 Upgrade Notes*—Use these upgrade notes to upgrade LabVIEW for Windows, Mac OS, and Linux to the latest version. The upgrade notes also describe new features and issues you might encounter when you upgrade.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

These documents are available in print and as PDFs in the `labview\manuals` directory. You must have Adobe Reader with Search and Accessibility 5.0.5 or later installed to view the PDFs. You must have Adobe Reader with Search and Accessibility 6.*x* or later installed to search PDF versions of these manuals. **(Mac OS)** You must have Adobe Reader with Search and Accessibility 6.*x* or later installed to view the PDFs.

Refer to the Adobe Systems Incorporated Web site at `www adobe com` to download Acrobat Reader. Refer to the National Instruments Product Manuals Library at `ni com/manuals` for updated documentation resources.

Readme Documents

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

- *LabVIEW Readme*—Use this file to learn important last-minute information about LabVIEW, including installation and upgrade issues, compatibility issues, changes from the previous version of LabVIEW, and known issues with LabVIEW. Open the *LabVIEW Readme* by selecting Start»All Programs»National Instruments»LabVIEW 8.0»Readme and opening `readme html` or by navigating to the `labview\readme` directory and opening `readme html`.
- *LabVIEW Application Builder Readme*—Use this document to learn about installing the LabVIEW Application Builder, which is included in the LabVIEW Professional Development System and is available for purchase separately. Open the *LabVIEW Application Builder Readme* by selecting Start»All Programs»National Instruments»LabVIEW 8.0»Readme and opening `readme_AppBldr html` or by navigating to the `labview\readme` directory and opening `readme_AppBldr html`.

LabVIEW VI Templates, Example VIs, and Tools

Use the LabVIEW VI templates, example VIs, and tools as a starting point to help you design and build VIs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LabVIEW VI Templates

The built-in VI templates include the subVIs, functions, structures, and front panel objects you need to get started building common measurement applications. VI templates open as untitled VIs that you must save. Select File»New to display the New dialog box, which lists the built-in VI templates. You also can display the New dialog box by clicking the New link in the Getting Started window.

LabVIEW Example VIs

LabVIEW searches among hundreds of example VIs you can use and incorporate into VIs that you create. 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. Browse or search the example VIs with the NI Example Finder by selecting Help»Find Examples.

Refer to NI Developer Zone at `ni.com/zone` for additional example VIs.

You also can access examples using the Open example and Browse related examples buttons located at the bottom of certain VI and function reference topics in the *LabVIEW Help*. Click the Open example button to open the example VI to which the topic refers. Click the Browse related examples button to open the NI Example Finder and display related example VIs.

You also 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.

LabVIEW Tools for DAQ Configuration (Windows)

Use Measurement & Automation Explorer (MAX) to help you configure measurement devices. Select Tools»Measurement & Automation Explorer to launch MAX and configure National Instruments hardware and software. You install MAX from the National Instruments Device Drivers CD.

Refer to the Controlling Instruments book on the Contents tab in the *LabVIEW Help* for information about controlling other types of instruments.

Use the DAQ Assistant to graphically configure channels or common measurement tasks. The DAQ Assistant Express VI does not appear on the Functions palette unless you have NI-DAQmx installed. Refer to the *DAQ*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Getting Started Guide for more information about installing NI-DAQmx. You can access the DAQ Assistant in the following ways:

- Place the DAQ Assistant Express VI on the block diagram.
- Right-click a DAQmx global channel control and select New Channel (DAQ Assistant) from the shortcut menu. Right-click a DAQmx task name control and select New Task (DAQ Assistant) from the shortcut menu. Right-click a DAQmx scale name control and select New Scale (DAQ Assistant) from the shortcut menu.
- Launch Measurement & Automation Explorer and select Data Neighborhood or Scales from the Configuration tree. Click the Create New button. Configure an NI-DAQmx channel, task, or scale.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2

Introduction to Virtual Instruments

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.

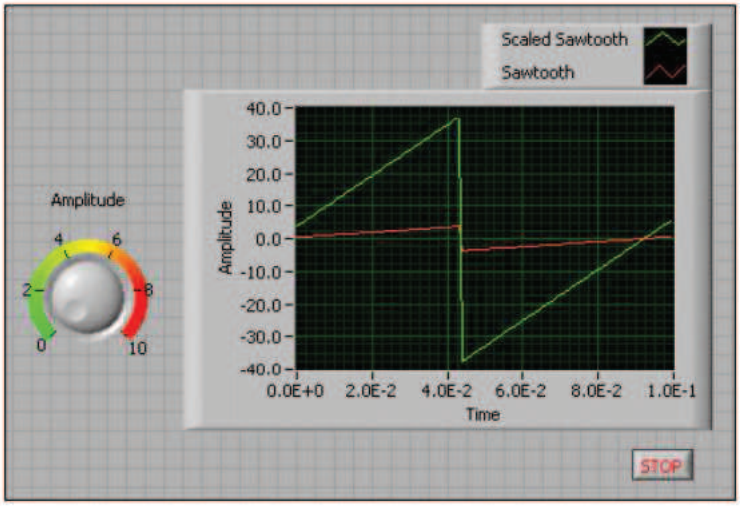
A VI contains the following three components:

- Front panel—Serves as the user interface.
- Block diagram—Contains the graphical source code that defines the functionality of the VI.
- Icon and connector pane—Identifies the interface to the VI so that 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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Front Panel

The front panel is the user interface of the VI. The following figure shows an example of a front panel.



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

Refer to Chapter 4, *Building the Front Panel*, for more information about the front panel.

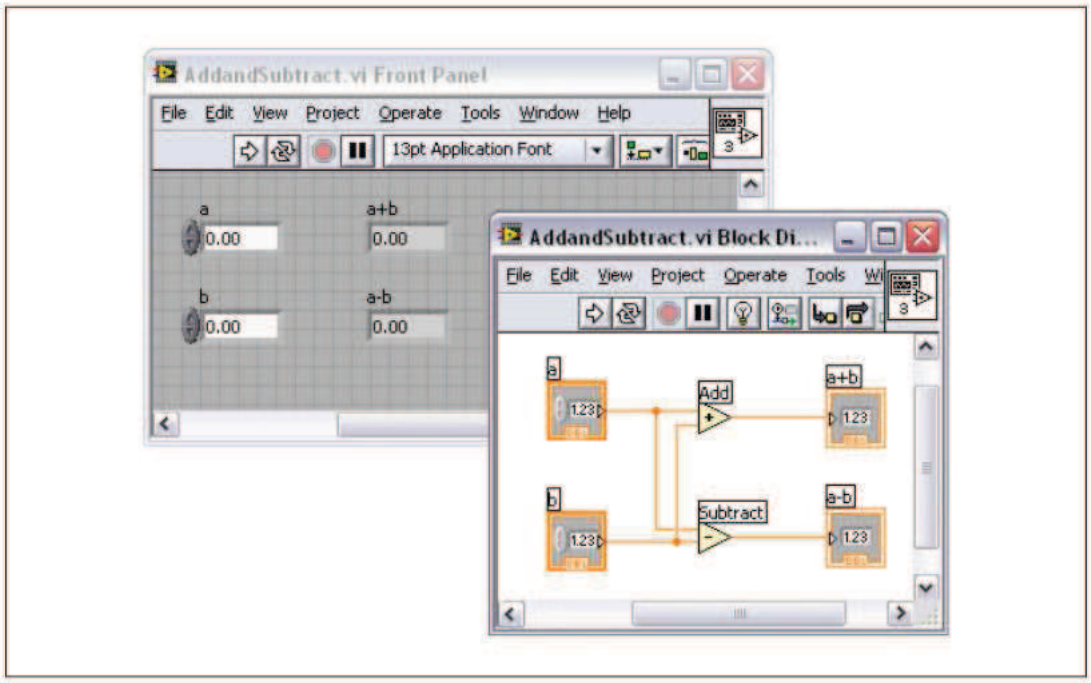
Block Diagram

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, also known as G code or block diagram code. Front panel objects appear as terminals on the block diagram.

Refer to Chapter 5, *Building the Block Diagram*, for more information about the block diagram.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following VI contains several primary block diagram objects—terminals, functions, and wires.



Terminals

The terminals represent the data type of the control or indicator. You can configure front panel controls or indicators to appear as icon or data type terminals on the block diagram. By default, front panel objects appear as icon terminals. For example, a knob icon terminal, shown as follows, represents a knob on the front panel.



The DBL at the bottom of the terminal represents a data type of double-precision, floating-point numeric. A DBL terminal, shown as follows, represents a double-precision, floating-point numeric control.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Refer to the *Control and Indicator Data Types* section of Chapter 5, *Building the Block Diagram*, for more information about data types in LabVIEW.

Terminals are entry and exit ports that exchange information between the front panel and block diagram. Data you enter into the front panel controls (a and b in the previous figure) enter the block diagram through the control terminals. The data then enter the Add and Subtract functions. When the Add and Subtract functions complete their calculations, they produce new data values. The data values flow to the indicator terminals, where they update the front panel indicators (a+b and a-b in the previous figure).

Nodes

Nodes are objects on the block diagram that have inputs and/or outputs and perform operations when a VI runs. They are analogous to statements, operators, functions, and subroutines in text-based programming languages. The Add and Subtract functions in the previous figure are examples of nodes.

Refer to the *Block Diagram Nodes* section of Chapter 5, *Building the Block Diagram*, for more information about nodes.

Wires

You transfer data among block diagram objects through wires. In the previous figure, wires connect the control and indicator terminals to the Add and Subtract functions. Each wire has a single data source, but you can wire it to many VIs and functions that read the data. Wires are different colors, styles, and thicknesses, depending on their data types. 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 when you try to wire two objects with incompatible data types.

Refer to the *Using Wires to Link Block Diagram Objects* section of Chapter 5, *Building the Block Diagram*, for more information about wires.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Structures

Structures are graphical representations of the loops and case statements of text-based programming languages. Use structures on the block diagram to repeat blocks of code and to execute code conditionally or in a specific order.

Refer to Chapter 8, *Loops and Structures*, for more information about structures.

Icon and Connector Pane

After you build a VI front panel and block diagram, build the icon and the connector pane so you can use the VI as a subVI. The icon and connector pane correspond to the function prototype in text-based programming languages. Every VI displays an icon, such as the one shown as follows, in the upper right corner of the front panel and block diagram windows.



An icon is a graphical representation of a VI. It can contain text, images, or a combination of both. If you use a VI as a subVI, the icon identifies the subVI on the block diagram of the VI. You can double-click the icon to customize or edit it.

Refer to the *Creating an Icon* section of Chapter 7, *Creating VIs and SubVIs*, for more information about icons.

You also need to build a connector pane, shown as follows, to use the VI as a subVI.



The connector pane is a set of terminals that correspond to the controls and indicators of that VI, similar to the parameter list of a function call in text-based programming languages. The connector pane defines the inputs and outputs you can wire to the VI so you can use it as a subVI. A connector pane receives data at its input terminals and passes the data to the block diagram code through the front panel controls and receives the results at its output terminals from the front panel indicators.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Refer to the *Building the Connector Pane* section of Chapter 7, *Creating VIs and SubVIs*, for more information about setting up connector panes.

Note Try not to assign more than 16 terminals to a VI. Too many terminals can reduce the readability and usability of the VI.

Using and Customizing VIs and SubVIs

After you build a VI and create its icon and connector pane, you can use it as a subVI.

Refer to the *Creating SubVIs* section of Chapter 7, *Creating VIs and SubVIs*, for more information about subVIs.

You can customize the appearance and behavior of a VI.

Refer to the *Customizing VIs* section of Chapter 7, *Creating VIs and SubVIs*, for more information about customizing a VI.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3

LabVIEW Environment

Use the LabVIEW palettes, tools, and menus to build the front panels and block diagrams of VIs. LabVIEW includes three palettes: the Controls palette, the Functions palette, and the Tools palette. LabVIEW also includes the Getting Started window, the Context Help window, the Project Explorer window, and the Navigation window. You can customize the Controls and Functions palettes, and you can set several work environment options.

Getting Started Window

The Getting Started window 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`.

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

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. The controls and indicators are located on subpalettes based on the types of controls and indicators.

Refer to the *Front Panel Controls and Indicators* section of Chapter 4, *Building the Front Panel*, for more information about the types of controls and indicators.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Select View»Controls Palette or right-click the front panel workspace to display the Controls palette. LabVIEW retains the Controls palette position and size so when you restart LabVIEW, the palette appears in the same position and has the same size. You can change the contents of the Controls palette.

Refer to the *Customizing the Controls and Functions Palettes* section of this chapter for more information about customizing the Controls palette.

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. The VIs and functions are located on subpalettes based on the types of VIs and functions.

Select View»Functions Palette or right-click the block diagram workspace to display the Functions palette. LabVIEW retains the Functions palette position and size so when you restart LabVIEW, the palette appears in the same position and has the same size. You can change the contents of the Functions palette.

Refer to the *Customizing the Controls and Functions Palettes* section of this chapter for more information about customizing the Functions palette.

Navigating the Controls and Functions Palettes

Click an object on the palette to place the object on the cursor so you can place it on the front panel or block diagram. You also can right-click a VI icon on the palette and select Open VI from the shortcut menu to open the VI.

Click the black arrows on the left side of the Controls or Functions palette to expand or collapse subpalettes. These arrows appear only if you set the palette format to Category (Standard) or Category (Icons and Text).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Use the following buttons on the Controls and Functions palette toolbars to navigate the palettes, to configure the palettes, and to search for controls, VIs, and functions.

	Up—Takes you up one level in the palette hierarchy. Click this button and hold the mouse button down to display a shortcut menu that lists each subpalette in the path to the current subpalette. Select a subpalette name in the shortcut menu to navigate to the subpalette. This button appears only if you set the palette format to Icons, Icons and Text, or Text.
	Search—Changes the palette to search mode so you can perform text-based searches to locate controls, VIs, or functions on the palettes. While a palette is in search mode, click the Return button to exit search mode and return to the palette.
	View—Provides options for selecting a format for the current palette, showing and hiding categories for all palettes, and sorting items in the Text and Tree formats alphabetically. Select Options from the shortcut menu to display the Controls/Functions Palettes page of the Options dialog box, in which you can select a format for all palettes. This button appears only if you click the thumbtack in the upper left corner of a palette to pin the palette.
	Restore Palette Size—Resizes the palette to its default size. This button appears only if you resize the Controls or Functions palette.

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. The cursor corresponds to the icon of the tool you select on the palette. Use the tools to operate and modify front panel and block diagram objects.

If automatic tool selection is enabled and you move the cursor over objects on the front panel or block diagram, LabVIEW automatically selects the corresponding tool from the Tools palette.

Select View»Tools Palette to display the Tools palette. LabVIEW retains the Tools palette position so when you restart LabVIEW, the palette appears in the same position.



Tip Press the <Shift> key and right-click to display a temporary version of the Tools palette at the location of the cursor.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Menus and Toolbars

Use the menu and toolbar items to operate and modify front panel and block diagram objects.

Menus

The menus at the top of a VI window contain items common to other applications, such as Open, Save, Copy, and Paste, and other items specific to LabVIEW. Some menu items also list keyboard shortcuts.

(Mac OS) The menus appear at the top of the screen.

(Windows and Linux) The menus display only the most recently used items by default. Click the arrows at the bottom of a menu to display all items. You can display all menu items by default by selecting Tools»Options, selecting Environment from the Category list, and removing the checkmark from the Use abridged menus checkbox.

Note Some menu items are unavailable while a VI runs.

Shortcut Menus

All LabVIEW objects have associated shortcut menus. As you create a VI, use the shortcut menu items to change the appearance or behavior of front panel and block diagram objects. To access the shortcut menu, right-click the object.

(Mac OS) Press <Command>-click to perform the same action as right-click.

Shortcut Menus in Run Mode

When a VI is running, or is in run mode, all front panel objects have an abridged set of shortcut menu items by default. Use the abridged shortcut menu items to cut, copy, or paste the contents of the object, to set the object to its default value, or to read the description of the object.

Some of the more complex controls have additional options. For example, the knob shortcut menu includes items to add a needle and to change the display of scale markers.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VI Toolbar

Use the buttons on the VI toolbar to run VIs, pause VIs, abort VIs, debug VIs, configure fonts, and align, group, and distribute objects.

Refer to Chapter 6, *Running and Debugging VIs*, for more information about some of the toolbar buttons, or refer to the *LabVIEW Help* for a complete list and descriptions of the toolbar buttons.

Project Explorer Window Toolbars

Use the buttons on the Standard, Project, Build, and Source Control toolbars to perform operations in a LabVIEW project. The toolbars are available at the top of the Project Explorer window. You might need to expand the Project Explorer window to view all of the toolbars.

Refer to the *Project Explorer Window* section of this chapter for more information about LabVIEW projects.

Context Help Window

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, constants, structures, palettes, properties, methods, events, dialog box components, and items in the Project Explorer window. You also can use the Context Help window to determine exactly where to connect wires to a VI or function.

Refer to the *Using Wires to Link Block Diagram Objects* section of Chapter 5, *Building the Block Diagram*, for more information about using the Context Help window to wire objects.

Select Help»Show Context Help to display the Context Help window. You also can display the Context Help window by clicking the Show Context Help Window button on the toolbar, shown as follows.



(Windows) You also can display the window by pressing the <Ctrl-H> keys. **(Mac OS)** Press the <Command-H> keys. **(Linux)** Press the <Alt-H> keys.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The Context Help window resizes to accommodate each object description. You also can resize the Context Help window to set its maximum size. LabVIEW retains the Context Help window position and size so when you restart LabVIEW, the window appears in the same position and has the same maximum size.

If a corresponding *LabVIEW Help* topic exists for an object the Context Help window describes, a blue Detailed help link appears in the Context Help window. Also, the Detailed help button in the Context Help window, shown as follows, is enabled. Click the link or the button to display more information about the object.



Project Explorer Window

Use the Project Explorer window to create and edit LabVIEW projects. Use projects to group together LabVIEW files and non-LabVIEW files, create build specifications, and deploy or download files to targets. Select File»New Project to display the Project Explorer window.

Navigation Window

The Navigation window displays an overview of the active front panel in edit mode or the active block diagram. Use the Navigation window to navigate large front panels or block diagrams. Click an area of the image in the Navigation window to display that area in the front panel or block diagram window. You also can click and drag the image in the Navigation window to scroll through the front panel or block diagram. Portions of the front panel or block diagram that are not visible appear dimmed in the Navigation window.

Select View»Navigation Window to display the Navigation window. **(Windows)** You also can display the window by pressing the <Ctrl-Shift-N> keys. **(Mac OS)** Press the <Command-Shift-N> keys. **(Linux)** Press the <Alt-Shift-N> keys.



Note The Navigation window is available only in the LabVIEW Full and Professional Development Systems.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Resize the Navigation window to resize the image it displays. LabVIEW retains the Navigation window position and size so when you restart LabVIEW, the window appears in the same position and has the same size.

Customizing Your Work Environment

You can customize the Controls and Functions palettes, and you can use the Options dialog box to select a palette format and set other work environment options.

Customizing the Controls and Functions Palettes

You can customize the Controls and Functions palettes in the following ways:

- Edit a palette set to rearrange the built-in palettes, create and move subpalettes, and so on using the Edit Controls and Functions Palette Set dialog box. Select Tools»Advanced»Edit Palette Set to display the Edit Controls and Functions Palette Set dialog box. Right-click the palette you want to modify and select from the options on the shortcut menu.
- Add items on the Functions palette to the Favorites category. On a pinned Functions palette, right-click an object and select Add Item to Favorites from the shortcut menu. In the Category (Standard) and Category (Icons and Text) formats, you also can expand a palette to display a subpalette, right-click the title of the subpalette, and select Add Subpalette to Favorites from the shortcut menu.

Setting Work Environment Options

Select Tools»Options to customize LabVIEW. Use the Options dialog box to set options for front panels, block diagrams, paths, performance and disk issues, the alignment grid, palettes, undo, debugging tools, colors, fonts, printing, the History window, and other LabVIEW features.

Use the Category list at the left side of the Options dialog box to select among the different categories of options.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4

Building the Front Panel

The front panel is the user interface of a VI. Generally, you design the front panel first and then design the block diagram to perform tasks on the inputs and outputs you create on the front panel.

Refer to Chapter 5, *Building the Block Diagram*, for more information about the block diagram.

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 mechanisms. Indicators are graphs, LEDs, and other output displays. Controls simulate instrument input mechanisms and supply data to the block diagram of the VI. Indicators simulate instrument output mechanisms and display data the block diagram acquires or generates.

Select View»Controls Palette to display the Controls palette and then select controls and indicators from the Controls palette and place them on the front panel.

Front Panel Controls and Indicators

Use the front panel controls and indicators located on the Controls palette to build the front panel. Types of controls and indicators include numeric controls and indicators such as slides and knobs, graphs, charts, Boolean controls and indicators such as buttons and switches, strings, paths, arrays, clusters, listboxes, tree controls, tables, ring controls, enumerated type controls, containers, and so on.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Styles of Controls and Indicators

Front panel controls and indicators can appear in modern, classic, or system style.

Modern and Classic Controls and Indicators

Many front panel objects have a high-color appearance. Set the monitor to display at least 16-bit color for optimal appearance of the objects.

The controls and indicators located on the Modern palette also have corresponding low-color objects. Use the controls and indicators located on the Classic palette to create VIs for 256-color and 16-color monitor settings.

System Controls and Indicators

Use the system controls and indicators located on the System palette in dialog boxes you create. The system controls and indicators are designed specifically for use in dialog boxes and include ring and spin controls, numeric slides, progress bars, scroll bars, listboxes, tables, string and path controls, tab controls, tree controls, buttons, checkboxes, radio buttons, and an opaque label that automatically matches the background color of its parent. These controls differ from those that appear on the front panel only in terms of appearance. These controls appear in the colors you have set up for your system.

Because the system controls change appearance depending on which platform you run the VI, the appearance of controls in VIs you create is compatible on all LabVIEW platforms. When you run the VI on a different platform, the system controls adapt their color and appearance to match the standard dialog box controls for that platform.

Refer to the *Designing Dialog Boxes* section of this chapter for information about designing dialog boxes.

Numeric Displays, Slides, Scroll Bars, Knobs, Dials, and Time Stamps

Use the numeric objects located on the Numeric and Classic Numeric palettes to create slides, scroll bars, knobs, dials, and numeric displays. The palette also includes color boxes and a color ramp for setting color values and time stamps for setting time and date values. Use the numeric objects to enter and display numeric data.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Numeric Controls and Indicators

Numeric controls and indicators are the simplest way to enter and display numeric data. You can resize these front panel objects horizontally to accommodate more digits. Change the value of a numeric control using any of the following ways:

- Use the Operating tool or the Labeling tool to click inside the digital display window and enter numbers from the keyboard.
- Use the Operating tool to click the increment or decrement arrow buttons of a numeric control.
- Use the Operating tool or the Labeling tool to place the cursor to the right of the digit you want to change and press the up or down arrow keys.

By default, LabVIEW displays and stores numbers like a calculator. A numeric control or indicator displays up to six digits before automatically switching to exponential notation. You can configure the number of digits LabVIEW displays before switching to exponential notation by right-clicking the numeric object and selecting Format and Precision from the shortcut menu to display the Format and Precision page of the Numeric Properties dialog box.

Slide Controls and Indicators

Slide controls and indicators are numeric objects with a scale. The slide controls and indicators include vertical and horizontal slides, a tank, and a thermometer. Change the value of a slide control using either of the following ways:

- Use the Operating tool to click or drag the slider to a new position.
- Use the digital display to enter data just as you do for numeric controls and indicators.

Slide controls or indicators can display more than one value. Right-click the object and select Add Slider from the shortcut menu to add more sliders. The data type of a control with multiple sliders is a cluster that contains each of the numeric values.

Refer to the *Clusters* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about clusters.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Scroll Bar Controls and Indicators

Scroll bar controls, similar to slide controls, are numeric objects you can use to scroll data. The scroll bar controls include vertical and horizontal scroll bars. Change the value of a scroll bar by using the Operating tool to click or drag the square scroll box to a new position, by clicking the increment and decrement arrows, or by clicking the spaces between the scroll box and the arrows.

Rotary Controls and Indicators

The rotary controls and indicators include knobs, dials, gauges, and meters. The rotary objects operate similarly to the slide controls and indicators because they are numeric objects with a scale. Change the value of a rotary control using either of the following ways:

- Use the Operating tool to click or drag the needle to a new position.
- Use the digital display to enter data just as you do for numeric controls and indicators.

Rotary controls or indicators can display more than one value. Right-click the object and select Add Needle to add new needles. The data type of a control with multiple needles is a cluster that contains each of the numeric values.

Refer to the *Clusters* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about clusters.

Time Stamp Control and Indicator

Use the time stamp control and indicator to send and retrieve a time and date value to or from the block diagram. You can change the value of the time stamp control using any of the following ways:

- Right-click the control and select Format & Precision from the shortcut menu.
- Click the Time/Date Browse button, shown as follows, to display the Set Time and Date dialog box.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Right-click the control and select Data Operations»Set Time and Date from the shortcut menu to display the Set Time and Date dialog box.
- Right-click the control and select Data Operations»Set Time to Now from the shortcut menu.

Graphs and Charts

Use the graph controls and indicators on the Graph and Classic Graph palettes to plot numeric data in graph or chart form.

Refer to Chapter 10, *Graphs and Charts*, for more information about using graphs and charts in LabVIEW.

Buttons, Switches, and Lights

Use the Boolean controls and indicators located on the Boolean and Classic Boolean palettes to create buttons, switches, and lights. Use Boolean controls and indicators to enter and display Boolean (TRUE/FALSE) values. For example, if you are monitoring the temperature of an experiment, you can place a Boolean warning light on the front panel to indicate when the temperature exceeds a certain level.

Boolean controls have six types of mechanical action that allow you to customize Boolean objects to create front panels that more closely resemble the behavior of physical instruments. Use the shortcut menu to customize the appearance of Boolean objects and how they behave when you click them.

Radio Buttons Controls

Use the radio buttons control to give users a list of items from which they can select only one item at a time. If you want to give users the option to select none or one item, right-click the control and select Allow No Selection from the shortcut menu to place a checkmark next to the menu item.

Because the data type of a radio buttons control is an enumerated type, you can use the radio buttons control to select the cases of a Case structure.

Refer to the *Enumerated Type Controls* section of this chapter for more information about enumerated type controls. Refer to the *Case Structures* section of Chapter 8, *Loops and Structures*, for more information about Case structures.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Refer to the Radio Buttons Control VI and the Radio Buttons with Event Structure VI in the `labview\examples\general\controls\booleans.llb` for examples of using a radio buttons control.

Text Entry Boxes, Labels, and Path Displays

Use the string and path controls and indicators on the String & Path and Classic String & Path palettes to create text entry boxes and labels and to enter or return the location of a file or directory.

String Controls and Indicators

Use the Operating or Labeling tool to enter or edit text in a string control on the front panel. By default, new or changed text does not pass to the block diagram until you terminate the edit session. At run time, you terminate the edit session by clicking elsewhere on the panel, changing to a different window, clicking the Enter button on the toolbar, or pressing the <Enter> key on the numeric keypad. Pressing the <Enter> key on the keyboard enters a carriage return.

Right-click a string control or indicator to select a display type for the text in the control or indicator, such as password display or hex display.

Refer to the *Strings on the Front Panel* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about string display types.

Combo Box Controls

Use the combo box control to create a list of strings you can cycle through on the front panel. A combo box control is similar to a text or menu ring control. However, the value and data type of a combo box control are strings instead of numbers as with ring controls.

Refer to the *Ring Controls* section of this chapter for more information about ring controls.

Refer to the *Case Structures* section of Chapter 8, *Loops and Structures*, for more information about Case structures.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Path Controls and Indicators

Use path controls and indicators to enter or return the location of a file or directory. **(Windows and Mac OS)** You also can drag a path, folder, or file from Windows Explorer and place it in a path control if dropping is enabled during run time.

Path controls and indicators work similarly to string controls and indicators, but LabVIEW formats the path using the standard syntax for the platform you are using.

Array, Matrix, and Cluster Controls and Indicators

Use the array, matrix, and cluster controls and indicators located on the Array, Matrix & Cluster and Classic Array, Matrix & Cluster palettes to create arrays, matrices, and clusters of other controls and indicators. Arrays group data elements of the same type. Clusters group data elements of mixed types. Matrices group rows or columns of real or complex scalar data for some math operations, such as linear algebra operations.

Refer to the *Grouping Data with Arrays and Clusters* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about arrays and clusters.

Listboxes, Tree Controls, and Tables

Use the listbox controls located on the List & Table and Classic List & Table palettes to give users a list of items from which to select.

Listboxes

You can configure listboxes to accept single or multiple selections. Use the multicolumn listbox to display more information about each item, such as the size of the item and the date it was created.

Tree Controls

Use the tree control to give users a hierarchical list of items from which to select. You organize the items you enter in the tree control into groups of items, or nodes. Click the expand symbol next to a node to expand it and display all the items in that node. You also click the symbol next to the node to collapse the node.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note You can create and edit tree controls only in the LabVIEW Full and Professional Development Systems. If a VI contains a tree control, you can run the VI in all LabVIEW packages, but you cannot configure the control in the Base Package.

Refer to the Directory Hierarchy in Tree Control VI in the `labview\examples\general\controls\Directory Tree Control llb` for an example of using a tree control.

Tables

Use the table control to create a table on the front panel.

Refer to the *Tables* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about using table controls.

Ring and Enumerated Type Controls and Indicators

Use the ring and enumerated type controls and indicators located on the Ring & Enum and Classic Ring & Enum palettes to create a list of strings you can cycle through.

Ring Controls

Ring controls are numeric objects that associate numeric values with strings or pictures. Ring controls appear as pull-down menus that users can cycle through to make selections.

Ring controls are useful for selecting mutually exclusive items, such as trigger modes. For example, use a ring control for users to select from continuous, single, and external triggering.

Enumerated Type Controls

Use enumerated type controls to give users a list of items from which to select. An enumerated type control, or enum, is similar to a text or menu ring control. However, the data type of an enumerated type control includes information about the numeric values and the string labels in the control. The data type of a ring control is numeric.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Container Controls

Use the container controls located on the Containers and the Classic Containers palettes to group controls and indicators or to display the front panel of another VI on the front panel of the current VI. **(Windows)** You also can use container controls to display .NET and ActiveX objects on the front panel.

Refer to the *.NET and ActiveX Controls (Windows)* section of this chapter for more information about .NET and ActiveX controls.

Tab Controls

Use tab controls to overlap front panel controls and indicators in a smaller area. A tab control consists of pages and tabs. Place front panel objects on each page of a tab control and use the tab as the selector for displaying different pages.

Tab controls are useful when you have several front panel objects that are used together or during a specific phase of operation. For example, you might have a VI that requires the user to first configure several settings before a test can start, then allows the user to modify aspects of the test as it progresses, and finally allows the user to display and store only pertinent data.

On the block diagram, the tab control is an enumerated type control by default. Terminals for controls and indicators placed on the tab control appear as any other block diagram terminal.

Refer to the *Enumerated Type Controls* section of this chapter for more information about enumerated type controls.

Subpanel Controls

Use the subpanel control to display the front panel of another VI on the front panel of the current VI. For example, you can use a subpanel control to design a user interface that behaves like a wizard. Place the Back and Next buttons on the front panel of the top-level VI and use a subpanel control to load different front panels for each step of the wizard.

Note You can create and edit subpanel controls only in the LabVIEW Full and Professional Development Systems. If a VI contains a subpanel control, you can run the VI in all LabVIEW packages, but you cannot configure the control in the Base Package.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Refer to the `labview\examples\general\controls\subpanel llb` for examples of using subpanel controls.

I/O Name Controls and Indicators

Use the I/O name controls and indicators on the I/O and Classic I/O palettes to pass DAQ channel names, VISA resource names, and IVI logical names you configure to I/O VIs to communicate with an instrument or a DAQ device.

I/O name constants are located on the Functions palette. A constant is a terminal on the block diagram that supplies fixed data values to the block diagram.

Note All I/O name controls or constants are available on all platforms. This allows you to develop I/O VIs on any platform that can communicate with devices that are platform specific. However, if you try to run a VI with a platform-specific I/O control on a platform that does not support that device, you will receive an error.

(Windows) Use Measurement & Automation Explorer, available from the Tools menu, to configure DAQ channel names, VISA resource names, and IVI logical names.

(Mac OS and Linux) Use the configuration utilities for your instrument to configure VISA resource names and IVI logical names. Refer to the documentation for your instrument for more information about the configuration utilities.

Waveform Control

Use the waveform control to manipulate individual data elements of a waveform. The waveform control carries the data, start time, and delta *t* of a waveform.

Refer to the *Waveform Data Type* section of Chapter 10, *Graphs and Charts*, for more information about the waveform data type.

Digital Waveform Control

Use the digital waveform control to manipulate the individual elements of a digital waveform.

Refer to the *Digital Waveform Data Type* section of Chapter 10, *Graphs and Charts*, for more information about the digital waveform data type.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Digital Data Control

The digital data control displays digital data arranged in rows and columns. Use the digital data control to build digital waveforms or to display digital data extracted from a digital waveform. Wire the digital waveform data control to a digital data indicator to view the samples and signals of a digital waveform.

References to Objects or Applications

Use the reference number controls located on the Refnum and Classic Refnum palettes to work with files, directories, devices, and network connections. Use the control refnum to pass front panel object information to subVIs.

A reference number, or refnum, is a unique identifier for an object, such as a file, device, or network connection.When you open a file, device, or network connection, LabVIEW creates a refnum associated with that file, device, or network connection. All operations you perform on open files, devices, or network connections use the refnums to identify each object. Use a refnum control to pass a refnum into or out of a VI. For example, use a refnum control to modify the contents of the file that a refnum is referencing without closing and reopening the file.

Because a refnum is a temporary pointer to an open object, it is valid only for the period during which the object is open. If you close the object, LabVIEW disassociates the refnum with the object, and the refnum becomes obsolete. If you open the object again, LabVIEW creates a new refnum that is different from the first refnum. LabVIEW allocates memory for an object that is associated with a refnum. Close the refnum to release the object from memory.

LabVIEW remembers information associated with each refnum, such as the current location for reading from or writing to the object and the degree of user access, so you can perform concurrent but independent operations on a single object. If a VI opens an object multiple times, each open operation returns a different refnum. LabVIEW automatically closes refnums for you when a VI finishes running, but it is a good programming practice to close refnums as soon as you are finished with them to most efficiently use memory and other resources. Close refnums in the opposite order that you opened them. For example, if you obtain a refnum to object A and invoke a method on object A to obtain a refnum to object B, close the refnum to object B first and then close the refnum to object A.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

.NET and ActiveX Controls (Windows)

Use the .NET and ActiveX controls located on the .NET & ActiveX palette to manipulate common .NET or ActiveX controls. You can add additional .NET or ActiveX controls to this palette for later use. Select Tools»NET & ActiveX»Add .NET Controls to Palette or Tools»NET & ActiveX» Add ActiveX Controls to Palette to convert a set of .NET or ActiveX controls, respectively, to custom controls and add them to the .NET & ActiveX palette.

Note Creating and communicating with .NET objects requires the .NET Framework 1.1 Service Pack 1 or later. National Instruments also strongly recommends that you use .NET objects only in LabVIEW projects.

Configuring Front Panel Objects

Use Properties dialog boxes or shortcut menus to configure how controls and indicators appear or behave on the front panel. Use Properties dialog boxes when you want to configure a front panel control or indicator using a dialog box that includes context help and in which you can set several properties at the same time for an object. Use shortcut menus to quickly configure common control and indicator properties. The options available in Properties dialog boxes and shortcut menus differ for different front panel objects. Any option you set using a shortcut menu is reflected in the Properties dialog box, and any option you set using the Properties dialog box is reflected in the shortcut menu.

Right-click a control or indicator on the front panel and select Properties from the shortcut menu to access the Properties dialog box for that object. You cannot access Properties dialog boxes for a control or indicator while a VI runs.

You also can create a custom control or indicator to extend the available set of front panel objects. Right-click the control and select Advanced» Customize from the shortcut menu to customize a control or indicator. You can save a custom control or indicator you created in a directory or LLB and use the custom control or indicator on other front panels.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Showing and Hiding Optional Elements

Front panel controls and indicators have optional elements you can show or hide, such as labels, captions, and digital displays. Set the visible elements for the control or indicator on the Appearance page of the Properties dialog box for the front panel object. You also can set the visible elements by right-clicking an object, selecting Visible Items from the shortcut menu, and selecting among the available options.

Changing Controls to Indicators and Indicators to Controls

LabVIEW initially configures objects in the Controls palette as controls or indicators based on their typical use. For example, if you place a toggle switch on the front panel, it appears as a control because a toggle switch is usually an input mechanism. If you place an LED on the front panel, it appears as an indicator because an LED is usually an output device.

Some palettes contain a control and an indicator for the same type or class of object. For example, the Numeric palette contains a numeric control and a numeric indicator because you can have a numeric input or a numeric output.

You can change a control to an indicator by right-clicking the object and selecting Change to Indicator from the shortcut menu, and you can change an indicator to a control by right-clicking the object and selecting Change to Control from the shortcut menu.

Replacing Front Panel Objects

You can replace a front panel object with a different control or indicator. When you right-click an object and select Replace from the shortcut menu, a temporary Controls palette appears. Select a control or indicator from the temporary Controls palette to replace the current object on the front panel.

Configuring the Front Panel

You can customize the front panel by changing the color of front panel objects, by aligning and distributing front panel objects, and so on.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Coloring Objects

You can change the color of many objects but not all of them. You can change the color of most front panel objects and the front panel and block diagram workspaces. You cannot change the color of system controls and indicators because these objects appear in the colors you have set up for your system.

Use the Coloring tool to right-click an object or workspace to change the color of front panel objects or the front panel and block diagram workspaces. You also can change the default colors for some objects by selecting Tools»Options and selecting Colors from the Category list.

Color can distract the user from important information so use color logically, sparingly, and consistently, if at all.

Aligning and Distributing Objects

Select Edit»Enable Panel Grid Alignment to enable the grid alignment on the front panel and align objects as you place them. Select Edit»Disable Panel Grid Alignment to disable the grid alignment and use the visible grid to align objects manually. You also can press the <Ctrl-#> keys to enable or disable the grid alignment. On French keyboards, press the <Ctrl-"> keys.

(Mac OS) Press the <Command-*> keys. **(Linux)** Press the <Alt-#> keys.

You also can use the alignment grid on the block diagram.

Select Tools»Options and select Alignment Grid from the Category list to hide or customize the grid.

To align objects after you place them, select the objects and select the Align Objects pull-down menu on the toolbar or select Edit»Align Items. To space objects evenly, select the objects and select the Distribute Objects pull-down menu on the toolbar or select Edit»Distribute Items.

Grouping and Locking Objects

Use the Positioning tool to select the front panel objects you want to group and lock together. Click the Reorder button on the toolbar and select Group or Lock from the pull-down menu. Grouped objects maintain their relative arrangement and size when you use the Positioning tool to move

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

and resize them. Locked objects maintain their location on the front panel and you cannot delete them until you unlock them. You can set objects to be grouped and locked at the same time. Tools other than the Positioning tool work normally with grouped or locked objects.

Resizing Objects

You can change the size of most front panel objects. When you move the Positioning tool over a resizable object, resizing handles or circles appear at the points where you can resize the object. When you resize an object, the font size remains the same. Resizing a group of objects resizes all the objects within the group.

Some objects change size only horizontally or vertically when you resize them, such as digital numeric controls and indicators. Others keep the same proportions when you resize them, such as knobs. The Positioning cursor appears the same, but the dashed border that surrounds the object moves in only one direction.

You can manually restrict the growth direction when you resize an object. To restrict the growth vertically or horizontally or to maintain the current proportions of the object, press the <Shift> key while you click and drag the resizing handles or circles. To resize an object around its center point, press the <Ctrl> key while you click and drag the resizing handles or circles.

(Mac OS) Press the <Option> key. **(Linux)** Press the <Alt> key.

To resize multiple objects to the same size, select the objects and select the Resize Objects pull-down menu on the toolbar. You can resize all the selected objects to the width or height of the largest or smallest object, and you can resize all the selected objects to a specific size in pixels.

Adding Space to the Front Panel without Resizing the Window

You can add space to the front panel without resizing the window. To increase the space between crowded or tightly grouped objects, press the <Ctrl> key and use the Positioning tool to click the front panel workspace. While holding the key combination, drag out a region the size you want to insert.

(Mac OS) Press the <Option> key. **(Linux)** Press the <Alt> key.

A rectangle marked by a dashed border defines where space will be inserted. Release the mouse button and the key to add the space.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Labeling

Use labels to identify objects on the front panel and block diagram.

LabVIEW includes two kinds of labels—owned labels and free labels. Owned labels belong to and move with a particular object and annotate that object only. You can move an owned label independently, but when you move the object that owns the label, the label moves with the object. You can hide owned labels, but you cannot copy or delete them independently of their owners. You can display a separate owned label called a unit label for numeric controls and indicators by right-clicking the numeric control or indicator and selecting Visible Items»Unit Label from the shortcut menu.

Free labels are not attached to any object, and you can create, move, rotate, or delete them independently. Use them to annotate front panels and block diagrams.

Free labels are useful for documenting code on the block diagram and for listing user instructions on the front panel. Double-click an open space or use the Labeling tool to create free labels or to edit either type of label.

Text Characteristics

LabVIEW uses fonts already installed on your computer. Use the Text Settings pull-down menu on the toolbar to change the attributes of text.

The Text Settings pull-down menu contains the following built-in fonts:

- Application Font—Default font used for Controls and Functions palettes and text in new controls
- System Font—Used for menus
- Dialog Font—Used for text in dialog boxes

If you select objects or text before you make a selection from the Text Settings pull-down menu, the changes apply to everything you select. If you select nothing, the changes apply to the default font. Changing the default font does not change the font of existing labels. It affects only those labels you create from that point on.

When you transfer a VI that contains built-in fonts to another platform, the fonts correspond as closely as possible.

The Text Settings pull-down menu also has Size, Style, Justify, and Color submenu items.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Designing User Interfaces

If a VI serves as a user interface or a dialog box, front panel appearance and layout are important. Design the front panel so users can easily identify what actions to perform. You can design front panels that look similar to instruments or other devices.

Using Front Panel Controls and Indicators

Controls and indicators are the main components of the front panel. When you design the front panel, consider how users interact with the VI and group controls and indicators logically. If several controls are related, add a decorative border around them or put them in a cluster. Use the decorations located on the Decorations palette to group or separate objects on a front panel with boxes, lines, or arrows. These objects are for decoration only and do not display data.

Designing Dialog Boxes

Select File»VI Properties and select Window Appearance from the Category pull-down menu to hide the menu bar and scroll bars and to create VIs that look and behave like standard dialog boxes for each platform.

If a VI contains consecutive dialog boxes that appear in the same screen location, organize them so that the buttons in the first dialog box do not directly line up with the buttons in the next dialog box. Users might double-click a button in the first dialog box and unknowingly click a button in the subsequent dialog box.

Use the system controls located on the System palette in dialog boxes you create.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5

Building the 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, also known as G code or block diagram code.

Block Diagram Objects

Objects on the block diagram include terminals and nodes. You build block diagrams by connecting the objects with wires. The color and symbol of each terminal indicate the data type of the corresponding control or indicator. Constants are terminals on the block diagram that supply fixed data values to the block diagram.

Block Diagram Terminals

Front panel objects appear as terminals on the block diagram. Double-click a block diagram terminal to highlight the corresponding control or indicator on the front panel.

Terminals are entry and exit ports that exchange information between the front panel and block diagram. Data values you enter into the front panel controls enter the block diagram through the control terminals. During execution, the output data values flow to the indicator terminals, where they exit the block diagram, reenter the front panel, and appear in front panel indicators.

LabVIEW has control and indicator terminals, node terminals, constants, and specialized terminals on structures. You use wires to connect terminals and pass data to other terminals. Right-click a block diagram object and select Visible Items»Terminals from the shortcut menu to view the terminals. Right-click the object and select Visible Items»Terminals again to hide the terminals. This shortcut menu item is not available for expandable VIs and functions.

You can configure front panel controls or indicators to appear as icon or data type terminals on the block diagram. By default, front panel objects

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

appear as icon terminals. For example, a knob icon terminal, shown as follows, represents a knob control on the front panel.



The DBL at the bottom of the terminal represents a data type of double-precision, floating-point numeric. A DBL terminal, shown as follows, represents a double-precision, floating-point numeric control.



Right-click a terminal and remove the checkmark next to the View As Icon shortcut menu item to display the data type for the terminal. Use icon terminals to display the types of front panel objects on the block diagram, in addition to the data types of the front panel objects. Use data type terminals to conserve space on the block diagram.

Note Icon terminals are larger than data type terminals, so you might unintentionally obscure other block diagram objects when you convert a data type terminal to an icon terminal.

Control terminals have a thicker border than indicator terminals. Also, arrows appear on front panel terminals to indicate whether the terminal is a control or an indicator. An arrow appears on the right if the terminal is a control, and an arrow appears on the left if the terminal is an indicator.

Control and Indicator Data Types

Common control and indicator data types include floating-point numeric, integer numeric, time stamp, enumerated, Boolean, string, array, cluster, path, dynamic, waveform, refnum, and I/O name. Refer to the *LabVIEW Help* for the complete list of control and indicator data types with their symbols and uses.

The color and symbol of each terminal indicate the data type of the corresponding control or indicator. Many data types have a corresponding set of functions that can manipulate the data, such as the String functions on the String palette that correspond to the string data type.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Symbolic Numeric Values

Undefined or unexpected data invalidate all subsequent operations. Floating-point operations return the following two symbolic values that indicate faulty computations or meaningless results:

- `NaN` (not a number) represents a floating-point value that invalid operations produce, such as taking the square root of a negative number.
- `1nf` (infinity) represents a floating-point value outside of the range for that data type. For example, dividing 1 by zero produces `1nf`.

  LabVIEW can return `+1nf` or `-1nf`. `+1nf` indicates the largest value possible for the data type and `-1nf` indicates the smallest value possible for the data type.

LabVIEW does not check for overflow or underflow conditions on integer values.

Constants

Constants are terminals on the block diagram that supply fixed data values to the block diagram. Universal constants are constants with fixed values, such as pi (□) and infinity (□). User-defined constants are constants you define and edit before you run a VI.

Most constants are located at the bottom or top of their palettes.

Create a user-defined constant by right-clicking an input terminal of a VI or function and selecting Create»Constant from the shortcut menu.

Use the Operating or Labeling tool to click the constant and edit its value. If automatic tool selection is enabled, double-click the constant to switch to the Labeling tool and edit the value.

Block Diagram Nodes

Nodes are objects on the block diagram that have inputs and/or outputs and perform operations when a VI runs. They are analogous to statements, operators, functions, and subroutines in text-based programming languages. LabVIEW includes the following types of nodes:

- Functions—Built-in execution elements, comparable to an operator, function, or statement.
- SubVIs—VIs used on the block diagram of another VI, comparable to subroutines.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Refer to the *Creating SubVIs* section of Chapter 7, *Creating VIs and SubVIs*, for more information about using subVIs on the block diagram.

- Express VIs—SubVIs designed to aid in common measurement tasks. You configure an Express VI using a configuration dialog box.

  Refer to the *Express VIs* section of this chapter for more information about using Express VIs.

- Structures—Execution control elements, such as For Loops, While Loops, Case structures, Flat and Stacked Sequence structures, Timed structures, and Event structures.

  Refer to Chapter 8, *Loops and Structures*, for more information about using structures.

Refer to the *LabVIEW Help* for the complete list of block diagram nodes.

Polymorphic VIs and Functions

Polymorphic VIs and functions can adjust to input data of different data types. Most LabVIEW structures are polymorphic, as are some VIs and functions.

Functions are polymorphic to varying degrees—none, some, or all of their inputs can be polymorphic. Some function inputs accept numeric values or Boolean values. Some accept numeric values or strings. Some accept not only scalar numeric values, but also arrays of numeric values, clusters of numeric values, arrays of clusters of numeric values, and so on. Some accept only one-dimensional arrays, although the array elements can be of any type. Some functions accept all types of data, including complex numeric values.

Refer to the *Grouping Data with Arrays and Clusters* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about arrays and clusters.

Functions Overview

Functions are the essential operating elements of LabVIEW. Function icons on the Functions palette have pale yellow backgrounds and black foregrounds. Functions do not have front panels or block diagrams but do have connector panes. You cannot open or edit a function.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Adding Terminals to Functions

You can change the number of terminals for some functions. For example, to build an array with 10 elements, you must add 10 terminals to the Build Array function.

You can add terminals to functions by using the Positioning tool to drag the top or bottom borders of the function up or down, respectively. You also can use the Positioning tool to remove terminals from functions, but you cannot remove a terminal that is already wired. You must first delete the existing wire to remove the terminal.

Refer to the *Using Wires to Link Block Diagram Objects* section of this chapter for more information about wiring objects.

Built-in VIs and Functions

The Functions palette also includes the VIs that ship with LabVIEW. Use these VIs and functions as subVIs in an application to reduce development time. Click the View button on the Functions palette and select Always Visible Categories»Show All Categories from the shortcut menu to display all categories on the Functions palette.

Refer to the *Using Built-In VIs and Functions* section of Chapter 7, *Creating VIs and SubVIs*, for more information about using the built-in VIs and functions.

Refer to the *LabVIEW Help* for detailed information about all built-in VIs and functions.

Express VIs

Use the Express VIs for common measurement tasks. Express VIs are nodes that require minimal wiring because you configure them with dialog boxes. The inputs and outputs for the Express VI depend on how you configure the VI. Express VIs appear on the block diagram as expandable nodes with icons surrounded by a blue field.

Refer to the *Getting Started with LabVIEW* manual for more information about using Express VIs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Wires to Link Block Diagram Objects

You transfer data among block diagram objects through wires. Each wire has a single data source, but you can wire it to many VIs and functions that read the data, similar to passing required parameters in text-based programming languages. You must wire all required block diagram terminals. Otherwise, the VI is broken and will not run. Display the Context Help window to see which terminals a block diagram node requires. The labels of required terminals appear bold in the Context Help window.

Refer to the *Correcting Broken VIs* section of Chapter 6, *Running and Debugging VIs*, for more information about broken VIs.

Wire Appearance and Structure

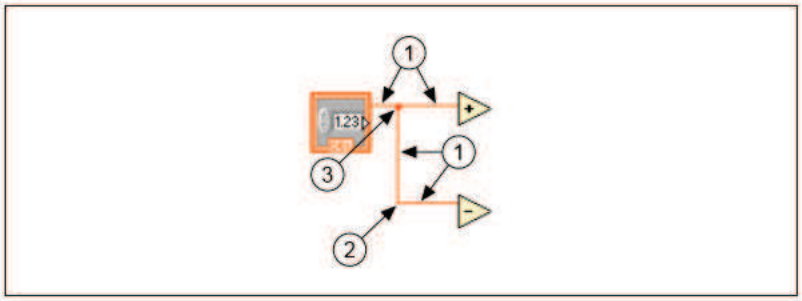
Wires are different colors, styles, and thicknesses depending on their data types, similar to how the color and symbol of a terminal indicate the data type of the corresponding control or indicator. A broken wire appears as a dashed black line with a red X in the middle. Broken wires occur for a variety of reasons, such as when you try to wire two objects with incompatible data types. The arrows on either side of the red X on the broken wire indicate the direction of the data flow, and the color of the arrows indicate the data type of the data flowing through the wire.

Refer to the *Control and Indicator Data Types* section of this chapter for more information about data types. Refer to the *Block Diagram Data Flow* section of this chapter for more information about data flow.

Wire stubs are the truncated wires that appear next to unwired terminals when you move the Wiring tool over a VI or function. They indicate the data type of each terminal. A tip strip also appears, listing the name of the terminal. After you wire a terminal, the wire stub for that terminal does not appear when you move the Wiring tool over its node.

A wire segment is a single horizontal or vertical piece of wire. A bend in a wire is where two segments join. The point at which two or more wire segments join is a junction. A wire branch contains all the wire segments from junction to junction, terminal to junction, or terminal to terminal if there are no junctions in between. The following figure shows a wire segment, bend, and junction.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



1 Segment 2 Bend 3 Junction

Wiring Objects

Use the Wiring tool to manually connect the terminals on one block diagram node to the terminals on another block diagram node. The cursor point of the tool is the tip of the unwound wire spool. When you move the Wiring tool over a terminal, the terminal blinks. When you move the Wiring tool over a VI or function terminal, a tip strip also appears, listing the name of the terminal. Wiring to the terminal might create a broken wire. You must correct the broken wire before you can run the VI.

Refer to the *Correcting Broken Wires* section of this chapter for more information about correcting broken wires.

Use the Context Help window to determine exactly where to connect wires. When you move the cursor over a VI or function, the Context Help window lists each terminal of the VI or function. The Context Help window does not display terminals for expandable VIs and functions, such as the Build Array function. Click the Show Optional Terminals and Full Path button in the Context Help window to display the optional terminals of the connector pane.

When you cross wires, a small gap appears in the first wire you drew to indicate that the first wire is under the second wire.

Bending Wires

While you are wiring a terminal, bend the wire at a 90 degree angle once by moving the cursor in either a vertical or horizontal direction. To bend a wire in multiple directions, click the mouse button to set the wire and then move the cursor in the new direction. You can repeatedly set the wire and move it in new directions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Undoing Wires

To undo the last point where you set the wire, press the <Shift> key and click anywhere on the block diagram. To abort the entire wiring operation, right-click anywhere on the block diagram.

(Mac OS) Press the <Option> key and click. **(Linux)** Click the middle mouse button.

Automatically Wiring Objects

As you move a selected object close to other objects on the block diagram, LabVIEW draws temporary wires to show you valid connections. When you release the mouse button to place the object on the block diagram, LabVIEW automatically connects the wires. You also can automatically wire objects already on the block diagram. LabVIEW connects the terminals that best match and does not connect the terminals that do not match.

Toggle automatic wiring by pressing the space bar while you move an object using the Positioning tool.

Selecting Wires

Select wires by using the Positioning tool to single-click, double-click, or triple-click them. Single-clicking a wire selects one segment of the wire. Double-clicking a wire selects a wire branch. Triple-clicking a wire selects the entire wire.

Correcting Broken Wires

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 when you try to wire two objects with incompatible data types. 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 to display more information about why the wire is broken.

Triple-click the wire with the Positioning tool and press the <Delete> key to remove a broken wire. You also can right-click the wire and select from shortcut menu options such as Delete Wire Branch, Create Wire Branch, Remove Loose Ends, Clean Up Wire, Change to Control, Change to

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

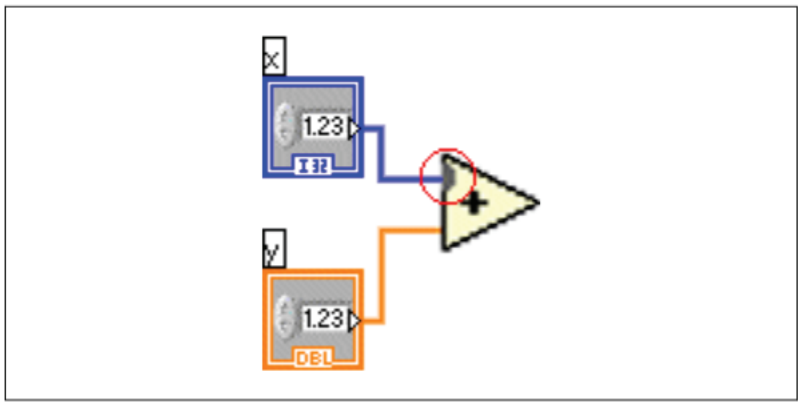
Indicator, Enable Indexing at Source, and Disable Indexing at Source. These options change depending on the reason for the broken wire.

You can remove all broken wires by selecting Edit»Remove Broken Wires or by pressing the <Ctrl-B> keys. **(Mac OS)** Press the <Command-B> keys. **(Linux)** Press the <Meta-B> keys.

Caution Use caution when removing all broken wires. Sometimes a wire appears broken because you are not finished wiring the block diagram.

Coercion Dots

Coercion dots appear on block diagram nodes to alert you that you wired two different numeric data types together. The dot means that LabVIEW converted the value passed into the node to a different representation. For example, the Add function expects two double-precision, floating-point inputs. If you change one of those inputs to an integer, a coercion dot appears on the Add function, as shown in the following figure.



Coercion dots can cause a VI to use more memory and increase its run time. Try to keep data types consistent in the VIs you create.

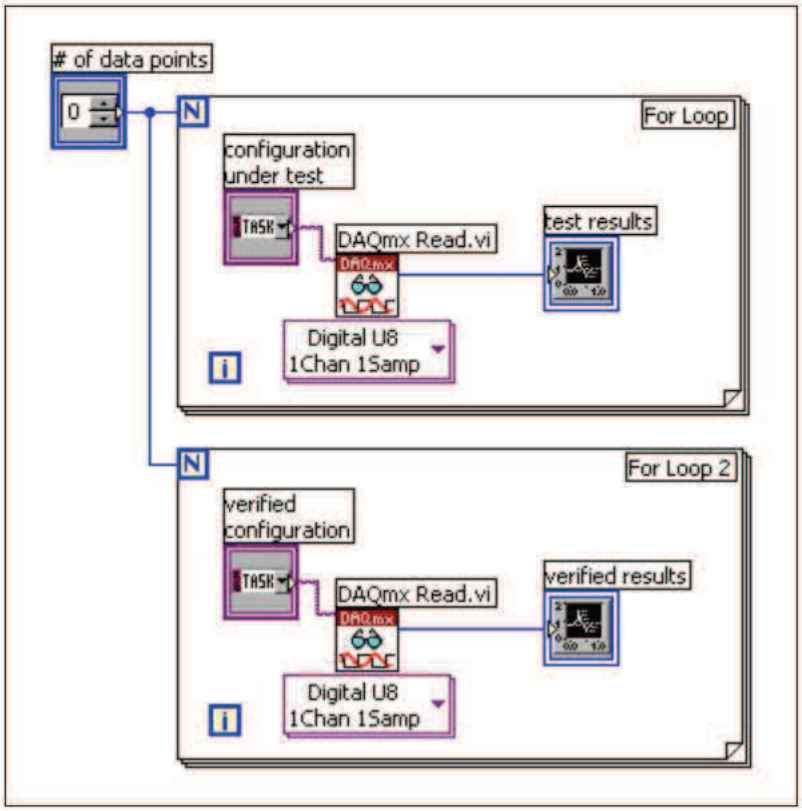
Block Diagram Data Flow

LabVIEW follows a dataflow model for running VIs. A block diagram node executes when it receives all required inputs. When a node executes, it produces output data and passes the data to the next node in the dataflow path. The movement of data through the nodes determines the execution order of the VIs and functions on the block diagram.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Visual Basic, C++, JAVA, and most other text-based programming languages follow a control flow model of program execution. In control flow, the sequential order of program elements determines the execution order of a program.

In LabVIEW, the flow of data rather than the sequential order of commands determines the execution order of block diagram elements. Therefore, you can create block diagrams that have simultaneous operations. For example, you can run two For Loops simultaneously and display the results on the front panel, as shown in the following block diagram.



Data Dependency and Artificial Data Dependency

The control flow model of execution is instruction driven. Dataflow execution is data driven, or data dependent. A node that receives data from another node always executes after the other node completes execution.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Block diagram nodes not connected by wires can execute in any order. You can use flow-through parameters to control execution order when natural data dependency does not exist. You can use a sequence structure to control execution order when flow-through parameters are not available.

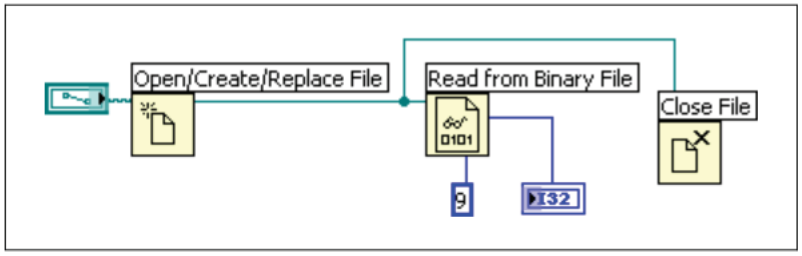
Refer to the *Flow-Through Parameters* section of this chapter for more information about flow-through parameters. Refer to the *Sequence Structures* section of Chapter 8, *Loops and Structures*, for more information about sequence structures.

You also can create an artificial data dependency, in which the receiving node does not actually use the data received. Instead, the receiving node uses the arrival of data to trigger its execution. Refer to the Timing Template (data dep) VI in the `labview\examples\general\structs llb` for an example of using artificial data dependency.

Missing Data Dependencies

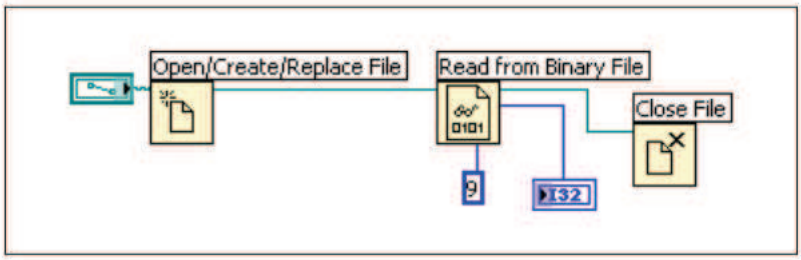
Do not assume left-to-right or top-to-bottom execution when no data dependency exists. Make sure you explicitly define the sequence of events when necessary by wiring the dataflow.

In the following block diagram, no dependency exists between the Read from Binary File function and the Close File function because the Read from Binary File function is not wired to the Close File function. This example might not work as expected because there is no way to determine which function runs first. If the Close File function runs first, the Read from Binary File function does not work.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following block diagram establishes a dependency by wiring an output of the Read from Binary File function to the Close File function. The Close File function does not run until it receives the output of the Read from Binary File function.



Flow-Through Parameters

Flow-through parameters, typically a refnum or error cluster, return the same value as the corresponding input parameter. Use these parameters to control execution order when natural data dependency does not exist. By wiring the flow-through output of the first node you want to execute to the corresponding input of the next node you want to execute, you create an artificial data dependency. Without these flow-through parameters, you must use sequence structures to ensure that data operations take place in the order you want.

Refer to the *Handling Errors* section of Chapter 6, *Running and Debugging VIs*, for more information about error I/O. Refer to the *Sequence Structures* section of Chapter 8, *Loops and Structures*, for more information about sequence structures.

Data Flow and Managing Memory

Dataflow execution makes managing memory easier than the control flow model of execution. In LabVIEW, you do not allocate memory for variables or assign values to them. Instead, you create a block diagram with wires that represent the transition of data.

VIs and functions that generate data automatically allocate the memory for that data. When the VI or function no longer uses the data, LabVIEW deallocates the associated memory. When you add new data to an array or a string, LabVIEW allocates enough memory to manage the new data.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Designing the Block Diagram

Use the following guidelines to design block diagrams:

- Use a left-to-right and top-to-bottom layout. Although the positions of block diagram elements do not determine execution order, avoid wiring from right to left to keep the block diagram organized and easy to understand. Only wires and structures determine execution order.
- Avoid creating a block diagram that occupies more than one or two screens. If a block diagram becomes large and complex, it can be difficult to understand or debug.
- Decide if you can reuse some components of the block diagram in other VIs or if a section of the block diagram works together as a logical component. If so, divide the block diagram into subVIs that perform specific tasks. Using subVIs helps you manage changes and debug the block diagrams quickly.

  Refer to the *Creating SubVIs* section of Chapter 7, *Creating VIs and SubVIs*, for more information about subVIs.
- Use the error handling VIs, functions, and parameters to manage errors on the block diagram.

  Refer to the *Handling Errors* section of Chapter 6, *Running and Debugging VIs*, for more information about handling errors.
- Avoid wiring under a structure border or between overlapped objects, because LabVIEW might hide some segments of the resulting wire.
- Avoid placing objects on top of wires. Placing a terminal or icon on top of a wire makes it appear as if a connection exists when it does not.
- Use free labels to document code on the block diagram.

  Refer to the *Labeling* section of Chapter 4, *Building the Front Panel*, for more information about using free labels.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6

Running and Debugging VIs

To run a VI, you must wire all the subVIs, functions, and structures with the correct data types for the terminals. Sometimes a VI produces data or runs in a way you do not expect. You can use LabVIEW to identify problems with block diagram organization or with the data passing through the block diagram.

Running VIs

Running a VI executes the operation for which you designed the VI. You can run a VI if the Run button on the toolbar appears as a solid white arrow, shown as follows.



The solid white arrow also indicates you can use the VI as a subVI if you create a connector pane for the VI.

Refer to the *Building the Connector Pane* section of Chapter 7, *Creating VIs and SubVIs*, for more information about creating connector panes.

A VI runs when you click the Run or Run Continuously buttons or the single-stepping buttons on the block diagram toolbar. While the VI runs, the Run button changes to a darkened arrow, shown as follows, to indicate that the VI is running.



You cannot edit a VI while the VI runs.

Clicking the Run button runs the VI once. The VI stops when the VI completes its data flow. Clicking the Run Continuously button, shown as follows, runs the VI continuously until you stop it manually.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Clicking the single-stepping buttons runs the VI in incremental steps.

Refer to the *Single-Stepping* section of this chapter for more information about using the single-stepping buttons to debug a VI.

Correcting Broken VIs

If a VI does not run, it is a broken, or nonexecutable, VI. The Run button appears broken, shown as follows, when the VI you are creating or editing contains errors.



If the button still appears broken when you finish wiring the block diagram, the VI is broken and cannot run.

Finding Causes for Broken VIs

Warnings do not prevent you from running a VI. They are designed to help you avoid potential problems in VIs. Errors, however, can break a VI. You must resolve any errors before you can run the VI.

Click the broken Run button or select View»Error List to find out why a VI is broken. The Error list window lists all the errors. The Items with errors section lists the names of all items in memory, such as VIs and project libraries that have errors. If two or more items have the same name, this section shows the specific application instance for each item. The errors and warnings section lists the errors and warnings for the VI you select in the Items with errors section. The Details section describes the errors and in some cases recommends how to correct the errors. Click the Help button to display a topic in the *LabVIEW Help* that describes the error in detail and includes step-by-step instructions for correcting the error.

Click the Show Error button or double-click the error description to highlight the area on the block diagram or front panel that contains the error.

The toolbar includes the Warning button, shown as follows, if a VI includes a warning and you placed a checkmark in the Show Warnings checkbox in the Error list window.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Common Causes of Broken VIs

The following list contains common reasons why a VI is broken while you edit it:

- The block diagram contains a broken wire because of a mismatch of data types or a loose, unconnected end.

  Refer to the *Correcting Broken Wires* section of Chapter 5, *Building the Block Diagram*, for information about correcting broken wires.

- A required block diagram terminal is unwired.

  Refer to the *Using Wires to Link Block Diagram Objects* section of Chapter 5, *Building the Block Diagram*, for information about setting required inputs and outputs.

- A subVI is broken or you edited its connector pane after you placed its icon on the block diagram of the VI.

  Refer to the *Creating SubVIs* section of Chapter 7, *Creating VIs and SubVIs*, for information about subVIs.

Debugging Techniques

If a VI is not broken, but you get unexpected data, you can use several techniques to identify and correct problems with the VI or the block diagram data flow.

Execution Highlighting

View an animation of the execution of the block diagram by clicking the Highlight Execution button, shown as follows.



Execution highlighting shows the movement of data on the block diagram from one node to another using bubbles that move along the wires. Use execution highlighting in conjunction with single-stepping to see how data values move from node to node through a VI.

Note Execution highlighting greatly reduces the speed at which the VI runs.

If the error out cluster reports an error, the error value appears next to error out with a red border. If no error occurs, OK appears next to error out with a green border.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Refer to the *Error Clusters* section of this chapter for more information about error clusters.

Single-Stepping

Single-step through a VI to view each action of the VI on the block diagram as the VI runs. The single-stepping buttons, shown as follows, affect execution only in a VI or subVI in single-step mode.



Enter single-step mode by clicking the Step Over or Step Into button on the block diagram toolbar. Move the cursor over the Step Over, Step Into, or Step Out button to view a tip strip that describes the next step if you click that button. You can single-step through subVIs or run them normally.

If you single-step through a VI with execution highlighting on, an execution glyph, shown as follows, appears on the icons of the subVIs that are currently running.



Probe Tool

Use a generic probe to view the data that passes through a wire. Right-click a wire and select Custom Probe»Generic Probe from the shortcut menu to use the generic probe.

Breakpoints

Use the Breakpoint tool, shown as follows, to place a breakpoint on a VI, node, or wire on the block diagram and pause execution at that location.



When you set a breakpoint on a wire, execution pauses after data passes through the wire. Place a breakpoint on the block diagram to pause execution after all nodes on the block diagram execute.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When a VI pauses at a breakpoint, LabVIEW brings the block diagram to the front and uses a marquee to highlight the node or wire that contains the breakpoint. When you move the cursor over an existing breakpoint, the black area of the Breakpoint tool cursor appears white.

When you reach a breakpoint during execution, the VI pauses and the Pause button appears red. You can take the following actions:

- Single-step through execution using the single-stepping buttons.
- Probe wires to check intermediate values.
- Change values of front panel controls.
- Click the Pause button to continue running to the next breakpoint or until the VI finishes running.

LabVIEW saves breakpoints with a VI, but they are active only when you run the VI. You can view all breakpoints by selecting Operate»Breakpoints and clicking the Find button.

Handling Errors

No matter how confident you are in the VI you create, you cannot predict every problem a user can encounter. Without a mechanism to check for errors, you know only that the VI does not work properly. Error checking tells you why and where errors occur.

When you perform any kind of input and output (I/O), consider the possibility that errors might occur. Almost all I/O functions return error information. Include error checking in VIs, especially for I/O operations (file, serial, instrumentation, data acquisition, and communication), and provide a mechanism to handle errors appropriately.

By default, LabVIEW automatically handles any error when a VI runs by suspending execution, highlighting the subVI or function where the error occurred, and displaying an error dialog box.

To disable automatic error handling for the current VI, select File»VI Properties and select Execution from the Category pull-down menu. To disable automatic error handling for any new, blank VIs you create, select Tools»Options and select Block Diagram from the Category list. To disable automatic error handling for a subVI or function within a VI, wire its error out parameter to the error in parameter of another subVI or function or to an error out indicator.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You can choose other error handling methods. For example, if an I/O VI on the block diagram times out, you might not want the entire application to stop and display an error dialog box. You also might want the VI to retry for a certain period of time. In LabVIEW, you can make these error handling decisions on the block diagram of the VI.

Use the LabVIEW error handling VIs and functions on the Dialog & User Interface palette and the error in and error out parameters of most VIs and functions to manage errors. For example, if LabVIEW encounters an error, you can display the error message in different kinds of dialog boxes. Use error handling in conjunction with the debugging tools to find and manage errors.

VIs and functions return errors in one of two ways—with numeric error codes or with an error cluster. Typically, functions use numeric error codes, and VIs use an error cluster, usually with error inputs and outputs.

Error handling in LabVIEW follows the dataflow model. Just as data values flow through a VI, so can error information. Wire the error information from the beginning of the VI to the end. Include an error handler VI at the end of the VI to determine if the VI ran without errors. Use the error in and error out clusters in each VI you use or build to pass the error information through the VI. The error clusters are flow-through parameters.

Refer to the *Flow-Through Parameters* section of Chapter 5, *Building the Block Diagram*, for more information about flow-through parameters.

As the VI runs, LabVIEW tests for errors at each execution node. If LabVIEW does not find any errors, the node executes normally. If LabVIEW detects an error, the node passes the error to the next node without executing that part of the code. The next node does the same thing, and so on. At the end of the execution flow, LabVIEW reports the error.

Error Clusters

The error in and error out clusters include the following components of information:

- status is a Boolean value that reports TRUE if an error occurred.
- code is a 32-bit signed integer that identifies the error numerically. A nonzero error code coupled with a status of FALSE signals a warning rather than a error.
- source is a string that identifies where the error occurred.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Some VIs, functions, and structures that accept Boolean data also recognize an error cluster. For example, you can wire an error cluster to the Boolean inputs of the Select, Quit LabVIEW, or Stop functions. If an error occurs, the error cluster passes a TRUE value to the function.

Refer to the *Clusters* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about clusters.

Using While Loops for Error Handling

You can wire an error cluster to the conditional terminal of a While Loop to stop the iteration of the While Loop. When you wire the error cluster to the conditional terminal, only the TRUE or FALSE value of the status parameter of the error cluster is passed to the terminal. When an error occurs, the While Loop stops.

When an error cluster is wired to the conditional terminal, the shortcut menu items Stop if True and Continue if True change to Stop on Error and Continue while Error.

Refer to the *While Loops* section of Chapter 8, *Loops and Structures*, for more information about using While Loops.

Using Case Structures for Error Handling

When you wire an error cluster to the selector terminal of a Case structure, the case selector label displays two cases—`Error` and `No Error`—and the border of the Case structure changes color—red for `Error` and green for `No Error`. If an error occurs, the Case structure executes the `Error` subdiagram.

Refer to the *Case Structures* section of Chapter 8, *Loops and Structures*, for more information about using Case structures.

Use the SubVI with Error Handling template VI to create a VI with a Case structure for error handling.

Refer to the *LabVIEW VI Templates* section of Chapter 1, *Introduction to LabVIEW*, for more information about template VIs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7

Creating VIs and SubVIs

A VI can serve as a user interface or as an operation you use frequently. After you learn how to build a front panel and block diagram, you can create your own VIs and subVIs and customize these VIs.

Searching for Examples

Before you build a new VI, consider searching for an example VI that meets your needs by selecting Help»Find Examples to open the NI Example Finder. If you cannot find an appropriate example VI, open a template VI from the New dialog box and populate the template with built-in VIs and functions from the Functions palette.

Refer to the *LabVIEW VI Templates, Example VIs, and Tools* section of Chapter 1, *Introduction to LabVIEW*, for more information about example VIs and template VIs.

Using Built-In VIs and Functions

LabVIEW includes built-in VIs and functions to help you build specific applications, such as data acquisition VIs and functions, VIs that access other VIs, VIs that communicate with other applications, and so on. You can use these VIs as subVIs in an application to reduce development time. Before you build a new VI, consider searching the Functions palette for similar VIs and functions and using an existing VI as the starting point for the new VI.

Creating SubVIs

After you build a VI, you can use it in another VI. A VI called from the block diagram of another VI is called a subVI. You can reuse a subVI in other VIs. To create a subVI, you need to build a connector pane and create an icon.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A subVI node corresponds to a subroutine call in text-based programming languages. The node is not the subVI itself, just as a subroutine call statement in a program is not the subroutine itself. A block diagram that contains several identical subVI nodes calls the same subVI several times.

The subVI controls and indicators receive data from and return data to the block diagram of the calling VI. Click the Select a VI icon or text on the Functions palette, navigate to and double-click a VI, and place the VI on a block diagram to create a subVI call to that VI.

You can edit a subVI by using the Operating or Positioning tool to double-click the subVI on the block diagram. When you save the subVI, the changes affect all calls to the subVI, not just the current instance.

Creating an Icon

Every VI displays an icon, shown as follows, in the upper right corner of the front panel and block diagram windows.



An icon is a graphical representation of a VI. It can contain text, images, or a combination of both. If you use a VI as a subVI, the icon identifies the subVI on the block diagram of the VI.

The default icon contains a number that indicates how many new VIs you have opened since launching LabVIEW. Create custom icons to replace the default icon by right-clicking the icon in the upper right corner of the front panel or block diagram and selecting Edit Icon from the shortcut menu or by double-clicking the icon in the upper right corner of the front panel.

You also can drag a graphic from anywhere in your file system and drop it in the upper right corner of the front panel or block diagram. LabVIEW converts the graphic to a 32 × 32 pixel icon.

Refer to the National Instruments Web site at `ni com/info` and enter the info code `expnr7` for standard graphics to use in a VI icon.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Building the Connector Pane

To use a VI as a subVI, you need to build a connector pane, shown as follows.



The connector pane is a set of terminals that corresponds to the controls and indicators of that VI, similar to the parameter list of a function call in text-based programming languages. The connector pane defines the inputs and outputs you can wire to the VI so you can use it as a subVI. A connector pane receives data at its input terminals and passes the data to the block diagram code through the front panel controls and receives the results at its output terminals from the front panel indicators.

Define connections by assigning a front panel control or indicator to each of the connector pane terminals. To define a connector pane, right-click the icon in the upper right corner of the front panel and select Show Connector from the shortcut menu to display the connector pane. The connector pane appears in place of the icon. When you view the connector pane for the first time, you see a connector pattern. You can select a different pattern by right-clicking the connector pane and selecting Patterns from the shortcut menu.

Each rectangle on the connector pane represents a terminal. Use the rectangles to assign inputs and outputs. The default connector pane pattern is 4 □2 □2 □4. If you anticipate changes to the VI that would require a new input or output, keep the default connector pane pattern to leave extra terminals unassigned.

You can assign up to 28 terminals to a connector pane. If your front panel contains more than 28 controls and indicators that you want to use programmatically, group some of them into a cluster and assign the cluster to a terminal on the connector pane.

Refer to the *Clusters* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about grouping data using clusters.

Select a different terminal pattern for a VI by right-clicking the connector pane and selecting Patterns from the shortcut menu. For example, you can select a connector pane pattern with extra terminals. You can leave the extra

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

terminals unconnected until you need them. This flexibility enables you to make changes with minimal effect on the hierarchy of the VIs.

Creating SubVIs from Sections of a VI

Convert a section of a VI into a subVI by using the Positioning tool to select the section of the block diagram you want to reuse and selecting Edit»Create SubVI. An icon for the new subVI replaces the selected section of the block diagram. LabVIEW creates controls and indicators for the new subVI, automatically configures the connector pane based on the number of control and indicator terminals you selected, and wires the subVI to the existing wires.

Creating a subVI from a selection is convenient but still requires careful planning to create a logical hierarchy of VIs. Consider which objects to include in the selection and avoid changing the functionality of the resulting VI.

Designing SubVI Front Panels

Place the controls and indicators on the front panel as they appear in the connector pane. Place the controls on the left of the front panel and the indicators on the right. Place the error in clusters on the lower left of the front panel and the error out clusters on the lower right.

Refer to the *Building the Connector Pane* section of this chapter for more information about setting up a connector pane.

Viewing the Hierarchy of VIs

The VI Hierarchy window displays a graphical representation of all open LabVIEW projects and targets, as well as the calling hierarchy for all VIs in memory, including type definitions and global variables. Select View»VI Hierarchy to display the VI Hierarchy window. Use this window to view the subVIs and other nodes that make up the VIs in memory and to search the VI hierarchy.

Refer to the *Project Explorer Window* section of Chapter 3, *LabVIEW Environment*, for more information about LabVIEW projects.

The VI Hierarchy window displays a top-level icon to represent the main LabVIEW application instance, under which appear all open VIs that are not part of a project or are not part of the application instance for a project. If you add a project, the VI Hierarchy window also displays another

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

top-level icon to represent the project. Each target you add appears under the project.

As you move the cursor over objects in the VI Hierarchy window, LabVIEW displays the name of each VI in a tip strip. You can use the Positioning tool to drag a VI from the VI Hierarchy window to the block diagram to use the VI as a subVI in another VI. You also can select and copy a node or several nodes to the clipboard and paste them on other block diagrams. Double-click a VI in the VI Hierarchy window to display the front panel of that VI.

A VI that contains subVIs has an arrow button on its bottom border. Click this arrow button to show or hide subVIs. A red arrow button appears when all subVIs are hidden. A black arrow button appears when all subVIs are displayed.

Polymorphic VIs

Polymorphic VIs accept different data types for a single input or output terminal. A polymorphic VI is a collection of VIs with the same connector pane patterns. Each VI in the collection is an instance of the polymorphic VI.

For example, the Read Key VI is polymorphic. Its default value terminal accepts Boolean; double-precision, floating-point numeric; 32-bit signed integer numeric; path; string; or 32-bit unsigned integer numeric data.

For most polymorphic VIs, the data types you wire to the inputs of the polymorphic VI determine the instance to use. If the polymorphic VI does not contain an instance compatible with that data type, a broken wire appears. If the data types you wire to the polymorphic VI inputs do not determine the instance to use, you must select the instance manually. If you manually select an instance of a polymorphic VI, the VI no longer behaves as a polymorphic VI because it accepts and returns only the data types of the instance you select.

To select the instance manually, right-click the polymorphic VI, select Select Type from the shortcut menu, and select the instance to use. You also can use the Operating tool to click the polymorphic VI selector, shown as follows, and select an instance from the shortcut menu.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Right-click the polymorphic VI on the block diagram and select Visible Items»Polymorphic VI Selector from the shortcut menu to display the selector. To change the polymorphic VI to accept all the handled data types again, right-click the polymorphic VI and select Select Type»Automatic from the shortcut menu or use the Operating tool to click the polymorphic VI selector and select Automatic from the shortcut menu.

Build polymorphic VIs when you perform the same operation on different data types.

Note You can build and edit polymorphic VIs only in the LabVIEW Professional Development System.

For example, if you want to perform the same mathematical operation on a single-precision floating-point numeric, an array of numeric values, or a waveform, you could create three separate VIs—Compute Number, Compute Array, and Compute Waveform. When you need to perform the operation, you place one of these VIs on the block diagram, depending on the data type you use as an input.

Instead of manually placing a version of the VI on the block diagram, you can create and use a single polymorphic VI.

Saving VIs

Select File»Save to save a VI. When you save a VI, you should use a descriptive name so you can easily identify the VI later. You also can save VIs for a previous version of LabVIEW to make upgrading LabVIEW convenient and to help you maintain the VIs in two versions of LabVIEW when necessary.

Naming VIs

When you save VIs, use descriptive names. Descriptive names, such as `Temperature Monitor vi` and `Serial Write & Read vi`, make it easy to identify a VI and know how you use it. If you use ambiguous names, such as `VI#1 vi`, you might find it difficult to identify VIs, especially if you have saved several VIs.

Consider whether your users will run the VIs on another platform. Avoid using characters that some operating systems reserve for special purposes, such as `\ /?*<>` and #.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note If you have several VIs of the same name saved on your computer, carefully organize the VIs in different directories or LLBs to avoid LabVIEW referencing the wrong subVI when running the top-level VI.

Saving for a Previous Version

You can save VIs for a previous version of LabVIEW to make upgrading LabVIEW convenient and to help you maintain the VIs in two versions of LabVIEW when necessary. Select File»Save For Previous Version to save for the previous version of LabVIEW.

When you save a VI for the previous version, LabVIEW converts not just that VI but all the VIs in its hierarchy, excluding files in the `labview\vi lib` directory.

Often a VI uses functionality not available in the previous version of LabVIEW. In such cases, LabVIEW saves as much of the VI as it can and produces a report of what it cannot convert. The report appears immediately in the Warnings dialog box. Click the OK button to acknowledge these warnings and close the dialog box. Click the Save to File button to save the warnings to a text file to review later.

Customizing VIs

You can configure VIs and subVIs to work according to your application needs. For example, if you plan to use a VI as a subVI that requires user input, configure the VI so that its front panel appears each time you call it.

Select File»VI Properties to configure the appearance and behavior of a VI. Use the Category pull-down menu at the top of the VI Properties dialog box to select from several different option categories.

The VI Properties dialog box includes the following option categories:

- General—Use this page to determine the current path where a VI is saved, its revision number, revision history, and any changes made since the VI was last saved. You also can use this page to edit the icon for the VI.
- Documentation—Use this page to add a description of the VI and link to a help file topic.

  Refer to the *Documenting VIs* section of Chapter 12, *Documenting and Printing VIs*, for more information about the documentation options.
- Security—Use this page to lock or password-protect a VI.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Window Appearance—Use this page to customize the window appearance of VIs, such as the window title and style.
- Window Size—Use this page to set the size of the window.
- Execution—Use this page to configure how a VI runs. For example, you can configure a VI to run immediately when it opens or to pause when called as a subVI.
- Editor Options—Use this page to set the size of the alignment grid for the current VI and to change the style of the control or indicator LabVIEW creates when you right-click a terminal and select Create»Control or Create»Indicator from the shortcut menu.

  Refer to the *Aligning and Distributing Objects* section of Chapter 4, *Building the Front Panel*, for more information about the alignment grid.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8

Loops and Structures

Structures are graphical representations of the loops and case statements of text-based programming languages. Use structures on the block diagram to repeat blocks of code and to execute code conditionally or in a specific order.

Like other nodes, structures have terminals that connect them to other block diagram nodes, execute automatically when input data are available, and supply data to output wires when execution completes.

Each structure has a distinctive, resizable border to enclose the section of the block diagram that executes according to the rules of the structure. The section of the block diagram inside the structure border is called a subdiagram. The terminals that feed data into and out of structures are called tunnels. A tunnel is a connection point on a structure border.

Use the following structures located on the Structures palette to control how a block diagram executes processes:

- For Loop—Executes a subdiagram a set number of times.
- While Loop—Executes a subdiagram until a condition occurs.
- Case structure—Contains multiple subdiagrams, only one of which executes depending on the input value passed to the structure.
- Sequence structure—Contains one or more subdiagrams that execute in sequential order.
- Event structure—Contains one or more subdiagrams that execute depending on how the user interacts with the VI.
- Timed Structures—Execute one or more subdiagrams with time bounds and delays.

Right-click the border of a structure to display its shortcut menu.

For Loop and While Loop Structures

Use the For Loop and the While Loop to control repetitive operations.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For Loops

A For Loop, shown as follows, executes a subdiagram a set number of times.



The value in the count terminal (an input terminal), shown as follows, indicates how many times to repeat the subdiagram.



Set the count explicitly by wiring a value from outside the loop to the left or top side of the count terminal, or set the count implicitly with auto-indexing.

Refer to the *Auto-Indexing to Set the For Loop Count* section of this chapter for more information about setting the count implicitly.

The iteration terminal (an output terminal), shown as follows, contains the number of completed iterations.



The iteration count always starts at zero. During the first iteration, the iteration terminal returns `0`.

Both the count and iteration terminals are 32-bit signed integers. If you wire a floating-point number to the count terminal, LabVIEW rounds it and coerces it to within range. If you wire `0` or a negative number to the count terminal, the loop does not execute and the outputs contain the default data for that data type.

Add shift registers to the For Loop to pass data from the current iteration to the next iteration.

Refer to the *Shift Registers* section of this chapter for more information about adding shift registers to a loop.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

While Loops

Similar to a Do Loop or a Repeat-Until Loop in text-based programming languages, a While Loop, shown as follows, executes a subdiagram until a condition occurs.



The While Loop executes the subdiagram until the conditional terminal, an input terminal, receives a specific Boolean value. The default behavior and appearance of the conditional terminal is Stop if True, shown as follows.



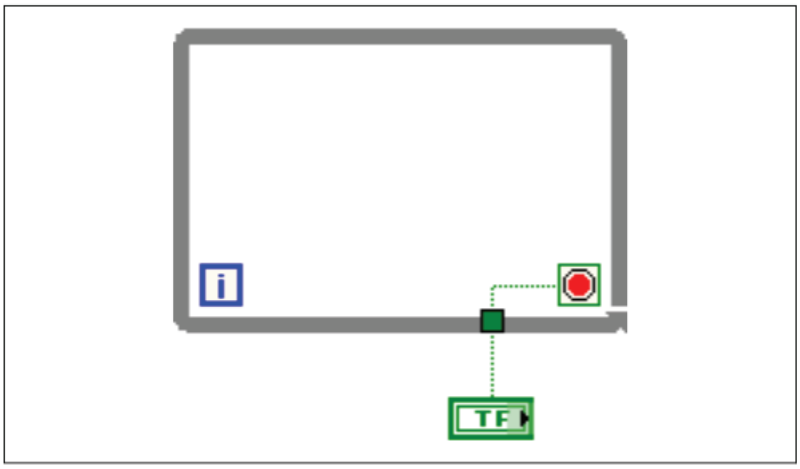
When a conditional terminal is Stop if True, the While Loop executes its subdiagram until the conditional terminal receives a TRUE value. You can change the behavior and appearance of the conditional terminal by right-clicking the terminal or the border of the While Loop and selecting Continue if True, shown as follows, from the shortcut menu.



When a conditional terminal is Continue if True, the While Loop executes its subdiagram until the conditional terminal receives a FALSE value. You also can use the Operating tool to click the conditional terminal to change the condition.

If you place the terminal of the Boolean control outside the While Loop, as shown in the following figure, and the control is set to FALSE if the conditional terminal is Stop if True when the loop starts, you cause an infinite loop. You also cause an infinite loop if the control outside the loop is set to TRUE and the conditional terminal is Continue if True.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Changing the value of the control does not stop the infinite loop because the value is only read once, before the loop starts. To stop an infinite loop, you must abort the VI by clicking the Abort Execution button on the toolbar.

You also can perform basic error handling using the conditional terminal of a While Loop. When you wire an error cluster to the conditional terminal, only the TRUE or FALSE value of the status parameter of the error cluster passes to the terminal. Also, the Stop if True and Continue if True shortcut menu items change to Stop if Error and Continue while Error.

Refer to the *Handling Errors* section of Chapter 6, *Running and Debugging VIs*, for more information about error clusters and error handling.

The iteration terminal (an output terminal), shown as follows, contains the number of completed iterations.



The iteration count always starts at zero. During the first iteration, the iteration terminal returns 0.

Add shift registers to the While Loop to pass data from the current iteration to the next iteration.

Refer to the *Shift Registers* section of this chapter for more information about adding shift registers to a loop.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP