

Preface

LabVIEW is a graphical programming language that has been widely adopted throughout industry, academia, and research labs as the standard for data acquisition and instrument control software. LabVIEW is a powerful and flexible instrumentation and analysis software system that is multiplatform—you can run LabVIEW on Windows, Mac OS X, and Linux. You can also run LabVIEW on PDAs (PalmOS, PocketPC, or Windows CE devices), on real-time platforms, and even embed LabVIEW programs into FPGA chips and 32-bit microprocessors. Creating your own LabVIEW program, or virtual instrument (VI), is simple. LabVIEW's intuitive user interface makes writing and using programs exciting and fun!

LabVIEW departs from the sequential nature of traditional programming languages and features an easy-to-use graphical programming environment, including all of the tools necessary for data acquisition (DAQ), data analysis, and presentation of results. With its graphical programming language, sometimes called "G," you program using a graphical block diagram that compiles into machine code. Ideal for a countless number of science and engineering applications, LabVIEW helps you solve many types of problems in only a fraction of the time and hassle it would take to write "conventional" code.

Beyond the Lab

LabVIEW has found its way into such a broad spectrum of virtual instrumentation applications that it is hard to know where to begin. As its name implies, it began in the laboratory and still remains very popular in many kinds of laboratories—from major research and development laboratories around the world (such as Lawrence Livermore, Argonne, Batelle, Sandia, Jet Propulsion Laboratory, White Sands, and Oak Ridge in the United States, and CERN in Europe), to R&D laboratories in many industries, and to teaching laboratories in universities all over the world, especially in the disciplines of electrical and mechanical engineering and physics.

The spread of LabVIEW beyond the laboratory has gone in many directions—up (aboard the space shuttle), down (aboard U.S. Navy submarines), and around the world (from oil wells in the North Sea to factories in New Zealand). And with the latest Internet capabilities, LabVIEW applications are being deployed not only physically in many places, but virtually across networked applications. More and more people are creating web-based control or monitoring of their LabVIEW applications to allow remote access and instant information about what's happening in their lab. Virtual instrumentation systems are known for their low cost, both in hardware and development time, and their great flexibility.

The Expanding World of Virtual Instrumentation

Perhaps the best way to describe the expansion (or perhaps explosion) of LabVIEW applications is to generalize it. There are niches in many industries where measurements of some kind are required—most often of temperature, whether it be in an oven, a refrigerator, a greenhouse, a clean room, or a vat of soup. Beyond temperature, users measure pressure, force, displacement, strain, pH, and so on, ad infinitum. Personal computers are used virtually everywhere. LabVIEW is the catalyst that links the PC with measuring things, not only because it makes it easy, but also because it brings along the ability to analyze what you have measured and display it and communicate it halfway around the world if you so choose.

After measuring and analyzing something, the next logical step often is to change (control) something based upon the results. For example, measure temperature and then turn on either a furnace or a chiller. Again, LabVIEW makes this easy to do; monitoring and control have become LabVIEW strengths. Sometimes it is direct monitoring and control, or it may be through communicating with a programmable logic controller (PLC) in what is commonly called supervisory control and data acquisition (SCADA).

The Results

A few of LabVIEW's many uses include the following:

- Simulating heart activity
- Controlling an ice cream-making process
- Detecting hydrogen gas leaks on the space shuttle
- Monitoring feeding patterns of baby ostriches
- Modeling power systems to analyze power quality
- Measuring physical effects of exercise in lab rats
- Controlling motion of servo and stepper motors
- Testing circuit boards in computers and other electronic devices
- Simulating motion in a virtual reality system
- Allowing remote navigation and feedback over the web of a helium-filled blimp
- Automatically generating cover sheets for your TPS reports

Objectives of This Book

LabVIEW for Everyone will help you get LabVIEW up and running quickly and easily, and will start you down the road to becoming an expert LabVIEW developer. The book offers additional examples and activities to demonstrate techniques, identifies other sources of information about LabVIEW, and features descriptions of cool LabVIEW applications. You are invited to open, inspect, use, and modify any of the programs on the accompanying CD-ROM. You can also get updates to the examples, activities, book errata, and other related resources and information at <http://labviewforeveryone.com>. The CD-ROM also includes the 30-day evaluation version of LabVIEW 8.0 for Windows, which allows you to do just about everything the commercial version does during the evaluation period. You can also always get the latest evaluation version of LabVIEW at <http://ni.com/labview>.

This book expects you to have basic knowledge of your computer's operating system. If you don't have much computer experience, you may want to spend a little time with your operating system manual and familiarize yourself with your computer. For example, you should know how to access menus, open and save files, make backup disks, and use a mouse. It also helps if you have some basic programming experience with other languages (C, Java, FORTRAN, etc.), but it is not necessary to know another programming language to use LabVIEW.

After reading this book and working through the exercises, you should be able to do the following, and much more, with the greatest of ease:

- Write LabVIEW programs, called virtual instruments, or VIs.
- Employ various debugging techniques.
- Manipulate both built-in LabVIEW functions and VIs.
- Create and save your own VIs so that you can use them as subVIs, or subroutines.
- Design custom graphical user interfaces (GUIs).
- Save your data in a file and display it on a graph or chart.
- Build applications that use General Purpose Interface Bus (GPIB) or serial instruments.
- Create applications that use plug-in DAQ boards.
- Use built-in analysis functions to process your data.
- Optimize the speed and performance of your LabVIEW programs.
- Employ advanced techniques such as state machines and event structures.
- Control your VIs and publish your data over the Internet or on the Web, using LabVIEW's features like its built-in Web server and remote panels.
- Use LabVIEW to create your instrumentation applications.

LabVIEW for Everyone helps you get started quickly with LabVIEW to develop your instrumentation and analysis applications. The book is divided into two main sections: *Fundamentals* and *Advanced Topics*.

The *Fundamentals* section contains nine chapters and teaches you the fundamentals of programming in LabVIEW. The *Advanced Topics* section contains eight chapters that further develop your skills and introduce helpful techniques and optimizing strategies. We suggest that you work through the beginning section to master the basics; then, if you're short on time, skip around to what you really want to learn in the advanced section.

In both sections, chapters have a special structure to facilitate learning, as follows:

- *Overview, goals, and key terms* describe the main ideas covered in that chapter.
- The main sections are a discussion of the featured topics.
- *Activities* reinforce the information presented in the discussion.
- *Wrap It Up!* summarizes important concepts and skills taught in the chapter.
- Additional activities in many chapters give you more practice with the new material.

Fundamentals

Chapter 1, “What in the World is LabVIEW?,” describes LabVIEW and introduces you to some of LabVIEW’s features and uses.

In Chapter 2, “Virtual Instrumentation: Hooking Your Computer Up to the Real World,” you will get an overview of virtual instrumentation: how data acquisition, GPIB, serial port communication, and data analysis are performed with LabVIEW.

In Chapter 3, “The LabVIEW Environment,” you will get acquainted with the LabVIEW environment, including the LabVIEW Project Explorer, the essential parts of a virtual instrument (or VI), the Help window, menus, tools, palettes, and subVIs.

In Chapters 4 and 5, “LabVIEW Foundations” and “Yet More Foundations,” you will become familiar with the basics of programming in LabVIEW—using controls and indicators (such as numerics, Booleans, and strings); wiring, creating, editing, debugging, and saving VIs; creating subVIs; and documenting your work. You will also begin to understand why LabVIEW is considered a dataflow programming language.

Chapter 6, “Controlling Program Execution with Structures,” describes the basic programming structures in LabVIEW: While Loops, For Loops, shift registers, Case Structures, Sequence Structures, and Formula Nodes. It also teaches you how to introduce timing into your programs. You will be introduced to easy-to-use frameworks that combine the While Loop and Case Structure to build scalable applications.

In Chapter 7, “LabVIEW’s Composite Data: Arrays and Clusters,” you will learn how to use two important data structures—arrays and clusters—in your programs. You will also explore LabVIEW’s built-in functions for manipulating arrays and clusters. It also teaches you about the error cluster and how to perform proper error handling in LabVIEW.

Chapter 8, “LabVIEW’s Exciting Visual Displays: Charts and Graphs,” details the variety of charts and graphs available in LabVIEW and teaches you how to use them for animated and informative data presentation. It also introduces the waveform, time stamp, and dynamic datatypes.

Chapter 9, “Exploring Strings and File I/O,” discusses string data types, string functions, and tables. You will learn how to parse strings like a pro, using the regular expression syntax. It also talks a little about how to save data in and read data from a file, using LabVIEW’s simple File I/O VIs.

Advanced Topics

Chapter 10, “Signal Measurement and Generation: Data Acquisition,” teaches you some of the theory behind signals, data acquisition, and analog and digital I/O. You will learn some hardware considerations, and you will find a valuable guide to many common acronyms used in instrumentation. Chapter 10 also discusses software

setup for data acquisition using the Measurement & Automation Explorer (MAX) utility and configuring NI-DAQmx devices.

Chapter 11, “Data Acquisition in LabVIEW,” then takes the basics of data acquisition you learned in Chapter 10 and teaches you how to apply them in LabVIEW using the DAQmx VIs. You’ll learn about the easy-to-use and powerful DAQmx tasks in LabVIEW, as well as work through some activities to read and write analog data, digital data. The chapter ends with some more advanced forms of data acquisition applications, such as streaming data to a file or doing a triggered acquisition.

Chapter 12, “Instrument Control in LabVIEW,” discusses how to use external instruments with LabVIEW. You’ll learn about GPIB, serial, Ethernet, and other kinds of instrument interfaces, and how to use the powerful VISA framework with LabVIEW to easily communicate with them.

Chapter 13, “Advanced LabVIEW Structures and Functions,” covers some invaluable features like local and global variables, property nodes, invoke nodes, and the powerful Event Structure for event-driven programming. You will learn about the State Machine and Queued Message Handler application frameworks, as well as the Messaging and Synchronization functions: Queues, Notifiers, Semaphores, Rendezvous, and Occurrences. It also introduces you to some more of LabVIEW’s cool GUI widgets like trees, subpanels, graphics and sound, and so on.

Chapter 14, “Advanced LabVIEW Data Concepts,” discusses some more file I/O, showing you how to work with binary files and configuration files, as well as advanced text file functions. You’ll see how LabVIEW can both read and generate external code modules such as DLLs and Shared Libraries.

Chapter 15, “Advanced LabVIEW Features,” shows you how to configure VI behavior and appearance using VI Setup options. You’ll learn about the powerful VI Server and how you can dynamically control LabVIEW, VIs, and controls by reference. It also introduces LabVIEW’s very useful tools such as the Find function and VI Hierarchy window.

Chapter 16, “Connectivity in LabVIEW,” covers connectivity features in LabVIEW such as printing to the web, controlling VIs remotely over the web, sharing data with Shared Variables, networking, and report generation.

In Chapter 17, “The Art of LabVIEW Programming,” you will learn about good LabVIEW style and some new tips, such as how to add a customized look to your applications by importing pictures and using the Control Editor. Chapter 17 describes some good programming techniques that you can use to make your programs run faster, use less memory, port more easily to other platforms, and behave more efficiently overall.

In Appendix A, “CD Contents,” you can find an outline and description of the files contained on the accompanying CD-ROM, as well as high-level instructions for installing the LabVIEW evaluation version and finding the examples and activities discussed in this book.

In Appendix B, “Add-on Toolkits for LabVIEW,” you will learn about add-on toolkits available from National Instruments and third-party companies to increase LabVIEW’s functionality.

Appendix C, “Open Source Tools for LabVIEW: OpenG,” introduces you to free (as in speech) software and the OpenG.org community that collaboratively develops add-on toolkits for LabVIEW.




In Appendix D, “LabVIEW Object-Oriented Programming,” you will be introduced to object-oriented programming techniques in LabVIEW, including a history of LabVIEW object-oriented programming and some perspectives on things to come.

Appendix E, “Resources for LabVIEW,” contains links to various LabVIEW resources such as user groups, discussion forums, and various other online LabVIEW resources.

In Appendix F, “LabVIEW Certification Exams,” you will learn about the LabVIEW developer certification exams, how to prepare for them, what to expect, and the benefits of certification.

You will find a glossary and index at the end of the book.

The following table describes the conventions used in this book.

bold	Bold text denotes VI names, function names, menus, menu items, and palettes. In addition, bold text denotes VI input and output parameters. For example, “Choose TCP Read from the TCP Functions palette.”
<i>italic</i>	Italic text denotes emphasis, a cross reference, or an introduction to a key term or concept. For example, “A <i>control reference</i> is an object that points to a LabVIEW control or indicator and can manage its properties.”
Courier	Courier type denotes text or characters that you enter using the keyboard. It also denotes files and paths. Sections of code, programming examples, syntax examples, and messages and responses that the computer automatically prints to the screen also appear in this font. For example, “In the text book, enter <code>c:\data\datafile.txt</code> as the filename.”
	Note. This icon marks information to which you should pay special attention.
	Watch Out! This icon flags a common pitfall or special information that you should be aware of in order to keep out of trouble.
	Hint. This icon calls your attention to useful tips and hints on how to do something efficiently.



When you see this icon, it means the topic being discussed is a Certified LabVIEW Developer (CLD) exam topic. If you want to prepare to become a CLD, pay attention! See Appendix F for detailed information about the certification program, what you can gain by becoming certified, how to study, and what to expect while taking the exams.



In Chapter 6, you will be introduced to LabVIEW's Express technologies, which provide configurable automatic programming for common tasks. This icon indicates the section refers to an Express Technology topic.

A Note about Paths

Different platforms have different conventions for specifying path names. For example, Windows paths take the form `X:\LABVIEW\MINE.LLB\BINGO.VI`. The same path on a MacOS classic system would be denoted `Hard Drive Name:LabVIEW:Mine.jib:Bingo.vi`. On Linux or OS X machines, it would be `/usr/labview/mine.llb/bingo.vi`. Rather than specifying a full path, this book will list the default path from the LabVIEW directory or folder when telling you where to find an example VI. To simplify notation, we will use the Windows standard to describe paths; if you use Mac OS or UNIX machines, please substitute colons or forward slashes where necessary.

What's New in the Third Edition

LabVIEW For Everyone was the first book published aimed at the beginner LabVIEW user in 1997 for LabVIEW 4.0; since then, Prentice Hall and other publishers have produced over a dozen LabVIEW books on specific topics. The second edition of *LabVIEW For Everyone* was updated for LabVIEW 6.1 and introduced some new topics.

This third edition has been completely revised and updated for LabVIEW 8.0. Two new chapters were added, and the number of pages has almost doubled! The changes from the second edition include the following:

- Inclusion of the new Express VIs and Express technology in LabVIEW
- Inclusion of the Certified LabVIEW Developer callouts to identify key sections that provide information that is tested on the CLD certification exam
- Autotool

- Static VI reference
- Call by Reference advanced options
- Event structure and event-driven programming
- Dynamic data
- Variants
- Type definitions
- Configuration (INI) file VIs
- Calling DLLs from LabVIEW
- Shared variables
- Custom probes
- Search and replace
- XML
- Pipes
- Error handling design patterns, tips, and tricks
- While Loop + case structure design patterns
- State machine and queued message handler design patterns
- Messaging and Synchronization using Queues, Notifiers, Semaphores, Rendezvous, and Occurrences
- Tree controls, tab controls, drag and drop, subpanels, scrollbars, and splitter bars
- Regular expressions
- Diagram disable and conditional disable structures
- Using NI-DAQmx and the LabVIEW DAQmx VIs
- LabVIEW project explorer
- Alignment grid
- Timed structures (timed loop and timed sequence structure)
- Advanced report generation
- Polymorphic VIs

LabVIEW Installation Instructions

If you own a licensed version of LabVIEW and need instructions on how to install it, please see the release notes that came with your software.

Otherwise, you can install a 30-day evaluation version of LabVIEW. Although we've included an evaluation version installer on this book's CD, we encourage you

to check <http://ni.com/labview> for the latest evaluation version of LabVIEW, available as a download. The evaluation version of LabVIEW included on this CD, version 8.0 for Windows, will allow you to work through all the examples in this book. Remember, the evaluation version stops working after 30 days. After that, you can activate your installed LabVIEW to a licensed version by purchasing a license from National Instruments at <http://ni.com>.

In addition, you will need to access the *EVERYONE* directory from the CD-ROM in the back of this book. It contains the activities in this book and their solutions. You may want to copy the *EVERYONE* directory onto your PC so you can save your activity work there as well. (You can also get updates to the examples, activities, book errata, and other related resources and information at <http://labviewforeveryone.com>.)

Purchasing LabVIEW

If you would like information on how to purchase LabVIEW, contact National Instruments:

National Instruments
11500 N Mopac Expwy
Austin, TX 78759-3504
Telephone: 888-280-7645
Fax: 512-683-8411
Email: info@ni.com.
Web: <http://www.ni.com>