

LabVIEW Core 2

Publish Date: Jan 21, 2015

Overview

This page describes the LabVIEW Core 2 training course offered through NI Training and Certification.

Register online at ni.com/training or contact us in one of the following ways

- call: (800) 433-3488
- fax: (512) 683-9300
- email: info@ni.com

Outside North America, contact your local NI Office. Worldwide Contact Info: ni.com/global.

Table of Contents

1. [Course Outline](#)
2. [LabVIEW Core 2 Course Outline](#)
3. [Suggested Next Courses](#)
4. [Suggested Certification](#)

1. Course Outline

The LabVIEW Core 2 course is an extension of the LabVIEW Core 1 course and teaches you to use common design patterns to successfully implement and distribute LabVIEW applications for research, engineering, and testing environments. Topics covered include programmatic control of your user interface, techniques to optimize reuse of existing code, use of file I/O functions, and tools to create executables and installers. This course directly links LabVIEW functionality to your application needs and provides a jump-start for application development.

Duration	<ul style="list-style-type: none">• Instructor-led Classroom: Two (2) Days• Instructor-led Virtual: Three (3) 4-hour sessions, plus homework
Audience	<ul style="list-style-type: none">• New users and users preparing to develop applications• LabVIEW Core 1 course attendees• Users and technical managers evaluating LabVIEW or NI Developer Suite in purchasing decisions• Users pursuing the Certified LabVIEW Associate Developer certification
Prerequisites	<ul style="list-style-type: none">• Experience with Microsoft Windows• LabVIEW Core 1 or equivalent experience
NI Products Used During the Course	<ul style="list-style-type: none">• LabVIEW Professional Development System• NI Data Acquisition device• BNC-2120

After attending this course, you will be able to:

- Use local variables to modify front panel controls or stop parallel loops
- Apply common design patterns that use queues and events
- Programmatically control user interface objects
- Evaluate file I/O formats and use them in applications
- Modify existing code for improved usability
- Prepare, build, debug, and deploy stand-alone applications

[<<Return to Course Details](#)

2. LabVIEW Core 2 Course Outline

Lesson	Overview	Topics
Using Variables	This lesson explains how to use local variables to modify front panel control values, stop parallel loops, and circumvent dataflow limitations.	<ul style="list-style-type: none">• Communicating between parallel loops• Using local and global variables• Writing to controls and reading from indicators• Understanding and avoiding race conditions
Communicating Data Between Parallel Loops	This lesson describes asynchronous communication techniques for creating code that is UI-driven and synchronizes data between parallel loops.	<ul style="list-style-type: none">• Using queues to pass buffered data between loops• Using notifiers to broadcast data to multiple loops
Implementing Design Patterns	This lesson introduces you to design patterns. You learn about the specific benefits and functionality of these design patterns and how they can be used to reduce development time. You learn two different categories of programming design patterns: single loop and multiple loops.	<ul style="list-style-type: none">• Using single loop design patterns—Including the state machine design patterns and functional global variables• Using multiple loop design patterns—including producer/consumer design patterns• Handling errors• Generating error codes and messages• Timing a design pattern
Controlling the User Interface	This lesson describes methods to control the attributes of front panel objects programmatically, such as temporarily disabling a control. You learn how to use VI Server to access the properties and methods of front panel objects.	<ul style="list-style-type: none">• VI Server architecture• Using property nodes• Using invoke nodes• Creating and using control references
	This lesson describes different file formats for collecting and	<ul style="list-style-type: none">• Comparing file formats• Creating file and folder paths

File I/O Techniques	storing data and how to select the appropriate file format for your applications. You practice implementing modular code that reads or writes measurement data.	<ul style="list-style-type: none"> • Writing and reading binary files • Working with multichannel text files with headers • Accessing Technical data management streaming (TDMS) files in LabVIEW and Excel
Improving an Existing VI	This lesson focuses on methods to refactor inherited LabVIEW code and how to maximize reuse of existing code. Refactoring is the process of redesigning software to make it more readable and maintainable without altering its observable behavior.	<ul style="list-style-type: none"> • Refactoring inherited code • Typical issues when refactoring code
Creating and Distributing Applications	This lesson describes the process of creating stand-alone executables and installers for LabVIEW applications. You will learn how to use the Application Builder in LabVIEW.	<ul style="list-style-type: none"> • Preparing the files • Creating build specifications • Creating and debugging an application • Creating an Installer

[<<Return to course details](#)

3. Suggested Next Courses

- LabVIEW Core 3
- LabVIEW Connectivity
- LabVIEW Performance
- Data Acquisition and Signal Conditioning
- Embedded Control and Monitoring Using LabVIEW
- Other hardware courses

4. Suggested Certification

- Certified LabVIEW Associate Developer Certification

[<<Return to course details](#)