

LabVIEW Intermediate II Course: Performance and Connectivity

To be scheduled starting February 2004.

Note: Starting February 2004, the LabVIEW Intermediate I and Intermediate II courses replace the LabVIEW Advanced: Performance and Communication course (P/N 910514-xx).

Overview

The LabVIEW Intermediate II course builds on the skills taught in LabVIEW Intermediate I and teaches you memory management and performance-enhancing techniques to maximize application performance. You also learn how to extend application functionality by leveraging the capabilities of other applications using connectivity technologies such as DLLs, ActiveX, and the Internet – further reducing development time. The LabVIEW Intermediate I and II courses are part of a series of courses designed to substantially build your proficiency with LabVIEW and, if you are pursuing certification, help you prepare for exams.

Duration – Two Days

Audience

- LabVIEW and NI Developer Suite users who need to maximize the performance of their LabVIEW application or extend the functionality by connecting to the network or other applications
- LabVIEW users pursuing the Certified LabVIEW Developer or Architect certification

Prerequisites

- LabVIEW Intermediate I or equivalent experience

NI Products Used During the Course

- LabVIEW Professional Development System Version 7

REGISTRATION

Register online at ni.com/training or call (800) 433-3488 • Fax: (512) 683-9300
info@ni.com

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

NI can also hold this class onsite at your facility.

Ordering Information 910721-xx

-01 NI Corporate or Branch, -11 Regional, -21 Onsite

After attending this course you will be able to:

- Improve the run-time speed and memory use of your application
- Understand how multithreading and priorities work in LabVIEW and how to optimize multithreaded tasks in VIs
- Utilize advanced data management and synchronization techniques
- Use LabVIEW with ActiveX and .NET
- Use event programming in LabVIEW
- Use DLLs with LabVIEW
- Programmatically control LabVIEW VIs and applications using VI Server
- Share data between applications using Datasocket
- Use TCP/IP VIs to communicate with applications on other computers connected across a network

Suggested Next Courses

- LabVIEW Advanced Application Development
- Data Acquisition and Signal Conditioning
- LabVIEW Instrument Control

LabVIEW Intermediate II: Performance and Connectivity Course Outline

Day 1

Improving Performance

This lesson reviews key topics from the LabVIEW Intermediate I course and continues with topics on how to maximize VI performance, including how to improve run-time speed and memory use. This lesson also describes some common errors and provides techniques to debug an application which does not behave as expected. Topics include:

- Understanding multitasking, multithreading, and multiprocessing
- Using threads and priorities to manage performance
- Multithreading issues
- Monitoring VI performance
- Reducing execution time
- Understanding and optimizing memory use

Event Programming

LabVIEW is a dataflow programming environment where the flow of data determines the execution order of block diagram elements. However, it is sometimes useful for specific code to execute when some action, or event, occurs. LabVIEW Intermediate I introduces the concept of event programming when architecting an application. This lesson extends event programming concepts so that you can understand and implement static, dynamic, user and ActiveX events. Topics include:

- Event-driven programming
- Programming with statically registered user interface events
- Programming with dynamically registered events
- Programmatically generating an event (user event)
- Avoiding common event structure mistakes
- About ActiveX events

Shared Libraries (DLLs)

A shared library is a library of executable functions or data that can be used by a Windows application. This lesson describes shared library and Dynamic Link Library (DLL) concepts and how you can use and create shared libraries/DLLs with LabVIEW. Topics include:

- Overview of shared libraries (DLLs)
- Accessing shared libraries in LabVIEW
- Debugging Call Library Function errors
- Creating shared libraries (DLLs) with LabVIEW
- About thread-safe DLLs

Day 2

VI Server

This lesson describes VI Server, which is a mechanism for controlling LabVIEW VIs and applications programmatically. It also allows you to control VIs and applications remotely over a TCP/IP network. Topics include:

- Understanding VI Server and relevant object-oriented terminology
- VI Server programming model
- Using VI Server VIs and functions
- Implementing remote communication using VI Server
- Dynamically calling and loading VIs
- Understanding and using strictly typed and weakly typed VI references

Windows Communication Platforms

ActiveX technology provides a standard model for inter-application communication that different programming languages can implement on different platforms. LabVIEW supports ActiveX automation and container technologies. This lesson describes ActiveX technology, which enhances the interactions between LabVIEW and other ActiveX-enabled applications, such as Excel. This lesson also introduces the Windows .NET communication platform. Topics include:

- The history of Windows communication platforms
- ActiveX features in LabVIEW
- Using LabVIEW as an ActiveX automation client
- Using ActiveX containers to display ActiveX controls and embedded documents on the front panel
- Using LabVIEW as an ActiveX automation server
- LabVIEW and .NET

DataSocket

This lesson describes the built-in DataSocket capabilities of LabVIEW. DataSocket is a programming tool that enables you to read, write, and share data between applications and/or different data sources and targets across a network. Topics include:

- Understanding DataSocket VIs and functions
- Managing client connections and setting up security using the DataSocket Manager
- Using attributes to send multiple data types on the same data items
- Using front panel DataSocket to read and write data directly to the DataSocket
- Using DataSocket for bi-directional communication
- Creating DataSocket VIs that read and write data from multiple sources, such as files and OPC servers

TCP/IP

This lesson describes the different protocols and methods that LabVIEW supports for transferring data to and from other applications, including TCP/IP and how you can use the LabVIEW TCP VI and functions to communicate with applications on other computers connected across a network. This lesson also introduces the client/server model which is the basis for most communication protocols. Topics include:

- Communication protocols such as TCP/IP
- Client/server programming model
- Using the TCP/IP VIs and functions