

LabWindows™/CVI™ Core 1 Course Manual

Course Software Version 2010
January 2011 Edition
Part Number 325668A-01

Copyright

© 1994–2011 National Instruments Corporation. All rights reserved.

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.

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

For components used in USI (Xerces C++, ICU, HDF5, b64, Stingray, and STLport), the following copyright stipulations apply. For a listing of the conditions and disclaimers, refer to either the `USICopyrights.chm` or the *Copyrights* topic in your software.

Xerces C++. This product includes software that was developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright 1999 The Apache Software Foundation. All rights reserved.

ICU. Copyright 1995–2009 International Business Machines Corporation and others. All rights reserved.

HDF5. 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.

b64. Copyright © 2004–2006, Matthew Wilson and Synesis Software. All Rights Reserved.

Stingray. This software includes Stingray software developed by the Rogue Wave Software division of Quovadx, Inc.
Copyright 1995–2006, Quovadx, Inc. All Rights Reserved.

STLport. Copyright 1999–2003 Boris Fomitchev

Trademarks

CVI, LabVIEW, National Instruments, NI, ni.com, the National Instruments corporate logo, and the Eagle logo are trademarks of National Instruments Corporation. Refer to the *Trademark Information* at ni.com/trademarks for other National Instruments trademarks.

FireWire® is the registered trademark of Apple Inc. The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries. 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/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at ni.com/patents.

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 43 662 457990-0, Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 5050 9800, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 358 (0) 9 725 72511, France 01 57 66 24 24, Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737, Italy 39 02 41309277, Japan 0120-527196, 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 22 328 90 10, Portugal 351 210 311 210, Russia 7 495 783 6851, Singapore 1800 226 5886, Slovenia 386 3 425 42 00, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151, Taiwan 886 02 2377 2222, Thailand 662 278 6777, Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

For further support information, refer to the *Additional Information and Resources* appendix. To comment on National Instruments documentation, refer to the National Instruments Web site at ni.com/info and enter the Info Code feedback.

Contents

Student Guide

A. LabWindows/CVI Training and Certification Sequence	v
B. Course Description	v
C. What You Need to Get Started	vi
D. Installing the Course Software	vii
E. Course Goals	vii
F. Course Conventions	viii

Lesson 1

Introduction to LabWindows/CVI

Introduction	1-1
A. LabWindows/CVI Environment	1-2
B. Function Panels	1-20
C. Data Types and Scope of Variables	1-29
D. Debugging	1-33

Lesson 2

Graphical User Interface

Introduction	2-1
A. Graphical User Interface (GUI)	2-2
B. Creating a Panel	2-5
C. Creating Controls	2-7
D. GUI Controls	2-16
E. User Interface Library	2-23
F. Controlling a GUI	2-25
G. Event-Driven Programming	2-34
H. CodeBuilder	2-41
I. Pop-Up Panels	2-44
J. Timer Controls	2-45
K. Strip Charts	2-46

Lesson 3

Instrument Drivers

Introduction	3-1
A. Instrument Drivers	3-2
B. Creating an Instrument Driver	3-7

Lesson 4**Distributing LabWindows/CVI Applications**

Introduction.....	4-1
A. Release Executables.....	4-2
B. Creating Release Executables.....	4-4
C. Distributing Executables.....	4-8

Lesson 5**Data Acquisition**

Introduction.....	5-1
A. DAQ System Overview and Configuration.....	5-2
B. Using Measurement & Automation Explorer (MAX).....	5-7
C. Analog Input Configuration Considerations.....	5-12
D. Single-Point Analog Input.....	5-16
E. Multiple-Point Acquisition.....	5-18
F. Continuous Acquisition.....	5-20
G. Single-Point Analog Output.....	5-22
H. Multiple-Point Voltage Output.....	5-24
I. Continuous Multiple Voltage Output.....	5-25
J. Digital Input and Output.....	5-27
K. Counters.....	5-33
L. DAQ Assistant.....	5-36

Lesson 6**Instrument Control**

Introduction.....	6-1
A. Introduction to Instrument Control.....	6-2
B. GPIB Communication.....	6-5
C. Virtual Instrument Software Architecture (VISA).....	6-12
D. Instrument I/O Assistant.....	6-18
E. Instrument Drivers.....	6-21
F. Serial Communication.....	6-29

Appendix A**Additional Information and Resources**

Lesson 4: Distributing LabWindows/CVI Applications

TOPICS

- A. Release Executables
- B. Creating Release Executables
- C. Distributing Executables



ni.com/training

Introduction

This lesson describes how to create and distribute a release executable. It also discusses the LabWindows/CVI Run-Time Engine, which contains all the library code needed to run release executables.

A. Release Executables

- Run-Time Engine
 - Run-Time Engine is a set of DLLs and other support files
 - Must be present on target machine
 - Needs to be installed only once
- Error Checking
 - Debugging is disabled in release executables
 - Check the return value from every function for errors



| ni.com/training

A. Release Executables

You can create true 32-bit and 64-bit Windows executables in LabWindows/CVI. The LabWindows/CVI run-time libraries come in DLL form. Executables you create in LabWindows/CVI and executables you create in an external compiler use the same DLLs. If you run more than one program at a time, Windows loads only one copy of the DLL.

Run-Time Engine

LabWindows/CVI uses a run-time engine to run compiled executables and DLLs. The CVI run-time engine is a set of DLLs and other support files that contain all the built-in library, memory, and program execution tools required to run a CVI application. You must include the LabWindows/CVI Run-Time Engine on any target computers on which you want to run your executable or DLL. You need only one installation of the CVI run-time engine on each target machine, though you might have several CVI applications.

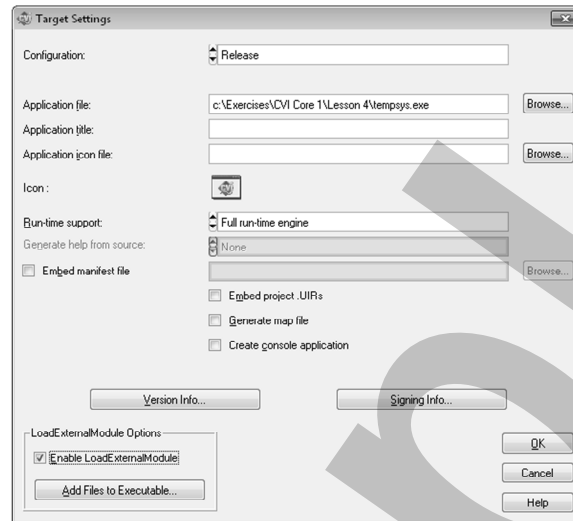
You receive the run-time engine with your purchase of LabWindows/CVI. The LabWindows/CVI installation program installs the files as part of the development environment. Use the **Build»Distributions** command in the Workspace window to bundle the run-time engine DLLs and drivers with your executable. You can distribute the run-time engine according to your license agreement. Code created with earlier versions of the run-time engine is fully compatible with the current version.

Error Checking

When you develop applications in LabWindows/CVI, you normally enable debugging. With the debugging option, LabWindows/CVI identifies and reports programming errors in your source code. In release executables, debugging is disabled, which results in smaller and faster executable code. However, you must do your own error checking because unpredictable results can occur. Remember that many problems can occur in release executables even if the program works inside the environment.

Check for errors that can occur due to external factors beyond the control of your program. Examples include running out of memory or trying to read from a file that does not exist. `malloc`, `fopen`, and `LoadPanel` are examples of functions that can encounter such errors.

B. Creating Release Executables



ni.com/training

B. Creating Release Executables

To create a release executable select a release configuration from the **Build»Configuration** menu, then select an executable from the **Build»Target Type** menu. Recall that release executables run without the debugger, and you must do your own error checking. If you try to debug a project in release mode, a message appears that asks if you want to run in debug mode instead. In release mode, select **Build»Create Release Executable**, then select **Run»Execute xxx.exe** to build and launch your application.

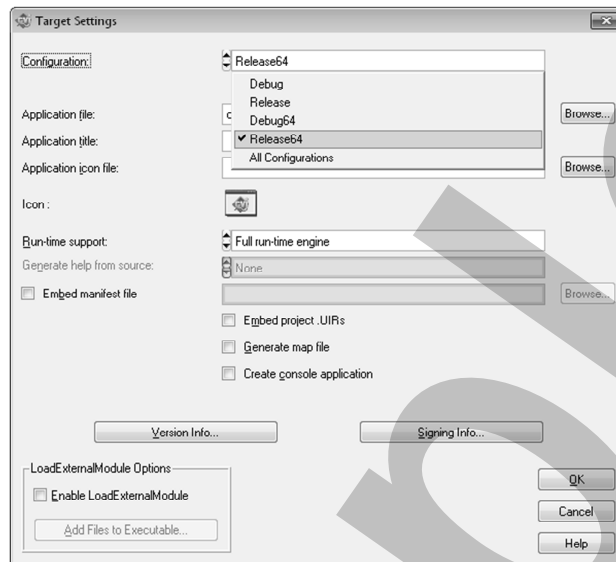
You can configure other advanced options for both the release and debug builds of your application in the Target Settings dialog box. Select **Build»Target Settings** to launch the Target Settings dialog box. You also can add version information, which can be retrieved by certain Windows SDK functions. You can find an example of adding version information in <National Instruments>\CVIversion\samples\sdk\verinfo.

If you enable the **Embed Project .UIRs** option in the Target Settings dialog box, LabWindows/CVI embeds .uir files into the project. Enabling this option allows you to ship only the executable, instead of including the .uir files also.

When you create your release executable, you can distribute your program to a target system. The remaining part of this lesson explains what files you need to transfer to the target system and how to use the automated installer feature to make the distribution easier.

To create an 32-bit executable that you can debug, select a debug configuration from the **Build»Configuration** menu, then select **Build»Create Debuggable Executable**.

64-bit Executables



ni.com/training

64-bit Executables

64-bit processors can handle arrays of larger sizes and can support more memory capacity than 32-bit processors. When you combine a 64-bit processor with a 64-bit operating system, you can take advantage of increased memory and data storage for your applications.



Note Throughout the *LabWindows/CVI Help* and within the LabWindows/CVI environment, the terms *64-bit platform* and *64-bit system* refer to a 64-bit processor running a 64-bit operating system.

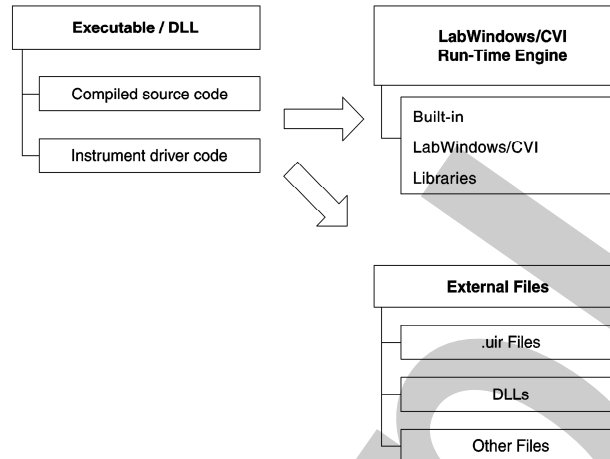
LabWindows/CVI includes configurations for building both 32-bit and 64-bit executables. This allows you to build a native 64-bit application as opposed to the default 32-bit application. Generally, writing code for 64-bit applications is no different than 32-bit applications. The primary difference is that you have access to more memory; the exact amount of memory depends on your processor. In 64-bit applications, pointers expand to 64 bits, meaning that you can address more memory than with 32-bit applications. Because the size of pointers is different between 32-bit applications, you can no longer assume that pointers are 32 bits long. You will need to update any code that makes this assumption.



Note You can run a 64-bit application only on a 64-bit system. However, you can compile 32- and 64-bit LabWindows/CVI projects on either a 32- or 64-bit computer.

Refer to the *Considerations for Porting Applications to 64-bit* and *Porting 32-Bit Code to 64-bit Code* topics of the *LabWindows/CVI Help* for more information about updating code to make it 64-bit ready.

Necessary Files



ni.com/training

Necessary Files

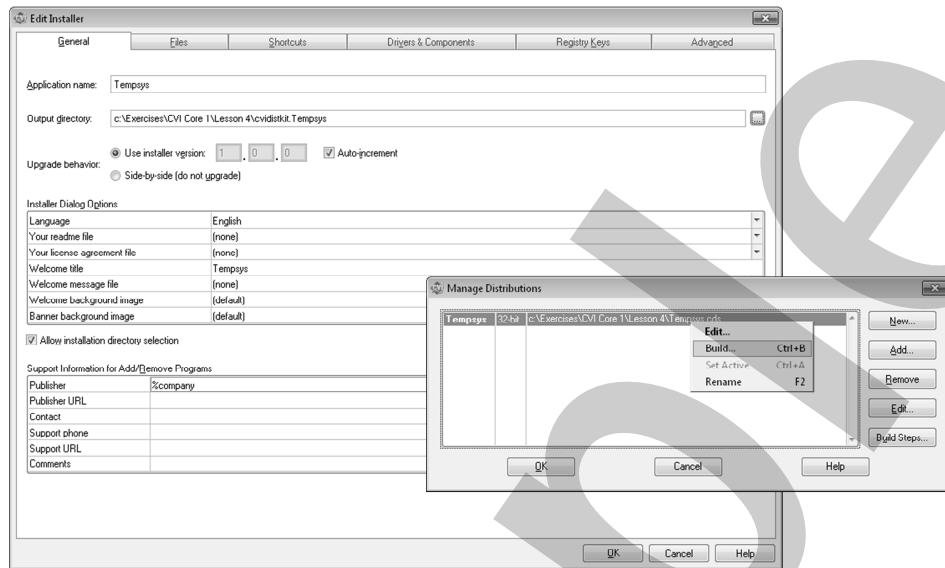
For your executable to run successfully on a target computer, make sure that all files required by the executable are accessible. If you create your distribution properly, the final distribution contains all the necessary files to install your LabWindows/CVI executable program on a target machine. Consider the following files:

- **Executable**—This file contains a compiled, linked version of your LabWindows/CVI project and the instrument driver program files that are statically linked to your project. The file also contains the application name and icon associated with the executable file. You can double-click the icon to start the application. When a user starts the executable, the executable loads and starts the run-time engine.
- **Run-Time Engine**—LabWindows/CVI Run-Time Engine contains all the built-in LabWindows/CVI libraries. The run-time engine consists of multiple files, including various DLLs and other support files.
- **.uir Files**—These files contain the user interface panels that the application uses. Use `LoadPanel`, `LoadPanelEx`, and `LoadMenuBar` to load these files. You do not need to include .uir files if you enable the **Embed project .UIRs** option in the Target Settings dialog box.
- **.dll Files**—You must include Windows DLL files used by your application program.

- **Other Files**—These files are additional files your program uses including, but not limited to, image files; files opened in your code using `open`, `fopen`, or `OpenFile`; and `.ini` files used for configuration.

Refer to the *Necessary Files for Running Executable Programs* topic of the *LabWindows/CVI Help* for a complete list of files to consider.

C. Distributing Executables



ni.com/training

C. Distributing Executables

After you create an executable, you can distribute it by creating a LabWindows/CVI distribution. A LabWindows/CVI distribution provides a complete way to bundle an application for installation and use on another computer. LabWindows/CVI distribution settings are saved in a .cds file, which is associated with a LabWindows/CVI workspace. LabWindows/CVI currently supports the Microsoft Windows Installer (.msi) distribution type. The Windows Installer technology packages all built libraries and executables along with any necessary files, drivers, and registry keys into a complete installer that can be run in a single operation on the target computer.

Select **Build»Distributions** to view a submenu you can use to build or edit the distributions in the current workspace. The **Distributions** submenu provides the following items:

- **Build Active**—Builds the active distribution for the workspace. Use the Manage Distributions dialog box to specify which distribution is the active distribution.
- **Edit Active**—Launches the Edit Installer dialog box for the active distribution. LabWindows/CVI adds the project output, project DLL dependencies, and any other project files, excluding .h, .c, .fp, and .lib files, to the installation. In the Edit Installer dialog box, use the **Drivers & Components** tab to add NI drivers or third-party modules, including the LabWindows/CVI Run-Time Engine, to the installation.

- **Manage Distributions**—Launches the Manage Distributions dialog box. Use this option to add, edit, remove, or create new distributions in the current workspace. In the Manage Distributions dialog box, right-click the distribution and select **Build** to build the installer.

The **Build** command automatically includes all the files necessary to run your executable program on a target computer except for DLLs that are loaded using `LoadLibrary`. Not all National Instruments hardware drivers are automatically included as part of your distribution. You must select these drivers from the **Drivers & Components** tab of the **Edit Installer** dialog box. For example, if your application uses a GPIB board, then you must install the DLLs that support your GPIB board by selecting the **NI-488.2** item in the **Drivers & Components** tab. Similarly, if your application uses a plug-in DAQ board, you must install the DLLs that will support your DAQ board by selecting the **NI-DAQmx** item in the **Drivers and Components** tab.

All stand-alone NI drivers or components are included as Windows installer files (.msi) from the original source distributions used to install the component on your system. You will be prompted to insert any necessary CDs during the distribution build process. Note that this may include more or different CDs than expected because of components included by dependency.



Note LabWindows/CVI displays the Distributions Needed dialog box when you build an installer that includes additional components that LabWindows/CVI cannot find.



Note LabWindows/CVI can also build 64-bit installers to install 64-bit applications. Choose 64-bit Windows Installer (.msi) from the Type dropdown when creating a new distribution to create a 64-bit installer.

Exercise 4-1 Creating a Distribution for an Executable

Create an executable from a project and create a distribution for the executable.

GOAL

Lesson 4 – Quiz

1. Which of the following types of files need to be included in a LabWindows/CVI distribution?
 - a. Source code files
 - b. DLL called by the application
 - c. Header files
 - d. Executable

2. Which of the following libraries is **not** available in the LabWindows/CVI Run-Time Engine?
 - a. User Interface Library
 - b. Formatting and I/O Library
 - c. NI driver libraries
 - d. Analysis/Advanced Analysis Library



| ni.com/training

Lesson 4 – Quiz Answers

1. Which of the following types of files need to be included in a LabWindows/CVI distribution (multiple answers)?
 - a. Source code files
 - b. DLL called by the application**
 - c. Header files
 - d. Executable**
2. Which of the following libraries is **not** available in the LabWindows/CVI Run-Time Engine?
 - a. User Interface Library
 - b. Formatting and I/O Library
 - c. NI driver libraries**
 - d. Analysis/Advanced Analysis Library



| ni.com/training

Lesson 4 – Quiz Answers

1. The current or later version of the LabWindows/CVI Run-Time engine DLL must exist on the target computer that you want to run your executable program. For example, if you built your executable with LabWindows/CVI 2010, the minimum version of the run-time engine must also be 2010.

The run-time engine includes built-in library, memory, and program execution tools. If your application uses a UIR file, you will need to distribute it with your executable or embed it in your executable.

2. The LabWindows/CVI Run-Time Engine does not include the shared libraries or drivers for National Instruments hardware. Users can install the shared libraries and drivers for their hardware from the distribution disks that National Instruments supplies.

Lesson 4 – Quiz

3. Can a 32-bit executable run in a 64-bit operating system?
- a. Yes
 - b. No



| ni.com/training

Lesson 4 – Quiz Answers

3. Can a 32-bit executable run in a 64-bit operating system?
 - a. Yes
 - b. No



| ni.com/training

Lesson 4 – Quiz Answers

3. The reverse is not true. A 64-bit executable cannot run in a 32-bit operating system.

Notes

Sample

Notes

Sample