



NATIONAL INSTRUMENTS™

**Measurement Studio™**

---

# Measurement Studio Evaluation Guide

## **Worldwide Technical Support and Product Information**

ni.com

## **National Instruments Corporate Headquarters**

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 794 0100

## **Worldwide Offices**

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011,  
Canada (Calgary) 403 274 9391, Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949,  
Canada (Québec) 514 694 8521, Canada (Toronto) 905 785 0085, China (Shanghai) 021 6555 7838,  
China (ShenZhen) 0755 3904939, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24,  
Germany 089 741 31 30, Greece 30 1 42 96 427, Hong Kong 2645 3186, India 91805275406,  
Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Malaysia 603 9596711,  
Mexico 5 280 7625, Netherlands 0348 433466, New Zealand 09 914 0488, Norway 32 27 73 00,  
Poland 0 22 528 94 06, Portugal 351 1 726 9011, Singapore 2265886, Spain 91 640 0085,  
Sweden 08 587 895 00, Switzerland 056 200 51 51, Taiwan 02 2528 7227, United Kingdom 01635 523545

For further support information, see the *Technical Support Resources* appendix. To comment on the documentation, send e-mail to [techpubs@ni.com](mailto:techpubs@ni.com).

Copyright © 2001 National Instruments Corporation. All rights reserved.

# 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 THEREOF 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.

## Trademarks

CodeBuilder™, ComponentWorks™, CVI™, DataSocket™, FlexMotion™, IMAQ™, IVI™, Measurement Studio™, National Instruments™, NI™, ni.com™, NI-488.2™, NI-DAQ™, NI-IMAQ™, and PXI™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

## 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 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.

# Conventions

---

The following conventions are used in this manual:

» 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 an activity that you can complete to practice the concepts presented in that section.



This icon denotes a tip, which alerts you to advisory information.



This icon denotes a note, which alerts you to important information.

**bold**

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

*italic*

Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font 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, functions, operations, variables, filenames and extensions, and code excerpts.

# Contents

---

## Chapter 1

### National Instruments, Virtual Instrumentation, and Measurement Studio

NI Platform .....	1-1
Virtual Instrumentation .....	1-2
Step 1—Acquire .....	1-2
Step 2—Analyze .....	1-2
Step 3—Present .....	1-2
Measurement Studio .....	1-4
Measurement Studio LabWindows/CVI .....	1-4
Measurement Studio for Visual C++ .....	1-4
Measurement Studio for Visual Basic .....	1-4
Related Software Packages .....	1-5
NI Developer Suite .....	1-5
TestStand .....	1-5
Vision and Image Processing Software .....	1-5
Motion Control Module .....	1-6
IVI Driver Toolset .....	1-6
LabWindows/CVI Enterprise Connectivity Toolset .....	1-6
Automation Symbols Toolset .....	1-7
PID Control Toolset .....	1-7

## Chapter 2

### Getting Started with Measurement Studio

About this Evaluation Package .....	2-1
Minimum System Requirements .....	2-2
Installation Instructions .....	2-3
Creating a New Project .....	2-3
LabWindows/CVI .....	2-3
Measurement Studio for Visual C++ .....	2-4
Measurement Studio for Visual Basic .....	2-4

## Chapter 3

### Exploring Measurement Studio

Interactive C Programming Environment .....	3-1
Microsoft Visual Studio Integration .....	3-2
Measurement Studio Wizards in Visual C++ .....	3-2
Measurement Studio Project Templates in Visual Basic .....	3-2
Measurement Libraries and Components .....	3-3
Data Acquisition and Instrument Control .....	3-3
Data Analysis .....	3-3
Presentation .....	3-4
Networking.....	3-4
Explore Measurement Libraries and Components.....	3-5
LabWindows/CVI.....	3-5
Measurement Studio for Visual C++ and Visual Basic.....	3-5
Configurable User Interface Controls .....	3-5
Create a Simple UI with a Graph and Buttons.....	3-6
LabWindows/CVI.....	3-6
Visual C++.....	3-7
Visual Basic.....	3-7
Context-Sensitive Help.....	3-7
LabWindows/CVI.....	3-8
Measurement Studio for Visual C++ .....	3-8
Measurement Studio for Visual Basic .....	3-8
LabWindows/CVI Code Generation .....	3-9
Simplified APIs and Native Data Types .....	3-10
Distribution.....	3-11

## Chapter 4

### Where to Go from Here

Tutorials and Related Documentation .....	4-1
Programmer and Reference Information .....	4-2
Measurement Studio Examples .....	4-3
Customer Education .....	4-3

## Appendix A

### Technical Support Resources

### Glossary

---

# National Instruments, Virtual Instrumentation, and Measurement Studio

National Instruments is committed to providing hardware and software for engineers, scientists, and systems integrators who build, maintain, and improve test, measurement, and automation applications. NI provides high performance, tight integration, and rapid application development of virtual instruments at a lower cost than traditional measurement instruments.

This chapter discusses the NI platform and provides an introduction to virtual instrumentation and NI Measurement Studio.

## NI Platform

---

NI pioneered the field of virtual instrumentation, which combines advanced hardware, powerful software, and off-the-shelf personal computers, to revolutionize the test and measurement field. This revolution has become the NI platform.

NI developed Measurement Studio to bridge the gap between standard software programming languages and virtual instrumentation needs. Measurement Studio provides functionality targeted at test, measurement, and automation applications for faster application development in the programming language of your choice—ANSI C, Visual C++, or Visual Basic.

Regardless of the environment you choose, Measurement Studio provides the language-specific tools to make your programming tasks easier. Because Measurement Studio provides all the tools in one package, you choose the language or combination of languages best suited for your application.

# Virtual Instrumentation

---

A virtual instrument is a combination of hardware and software in a PC that has the functionality of a classic stand-alone instrument and more. With a virtual instrument, you acquire and control data much like you do with traditional laboratory instruments. The power of NI virtual instrumentation comes with interactive data analysis, presentation, and distribution of your application.

Creating virtual instruments is a three-step process when you use NI hardware with Measurement Studio, regardless of the development environment you use—LabWindows/CVI, Visual C++, or Visual Basic.

## Step 1—Acquire

Acquire raw data through a hardware interface. Measurement Studio provides hardware connections through an extensive list of interfaces—PCI, PXI, GPIB, VXI, serial, and even Ethernet. Whether you collect data through a plug-in board or through a benchtop instrument, NI has the interface to collect the correct signals and measurements.

## Step 2—Analyze

Analyze raw data to make it meaningful to you and your users. Rather than collecting a list of raw signals and later analyzing them with a separate analysis application, use in-place analysis for fast and meaningful results. Measurement Studio offers libraries of analysis operations for all your measurement analysis needs.

## Step 3—Present

Present results in an intuitive and meaningful way and when applicable, allow users to interact with the instrumentation system through a front panel. A *front panel* is the interactive user interface of a virtual instrument.

Use three-dimensional, instrument-style controls and indicators—buttons, knobs, thermometers, tanks, LEDs, slides, numerical displays, strip charts, and graphs—to build front panels that represent traditional test and measurement instruments, as shown in Figure 1-1.



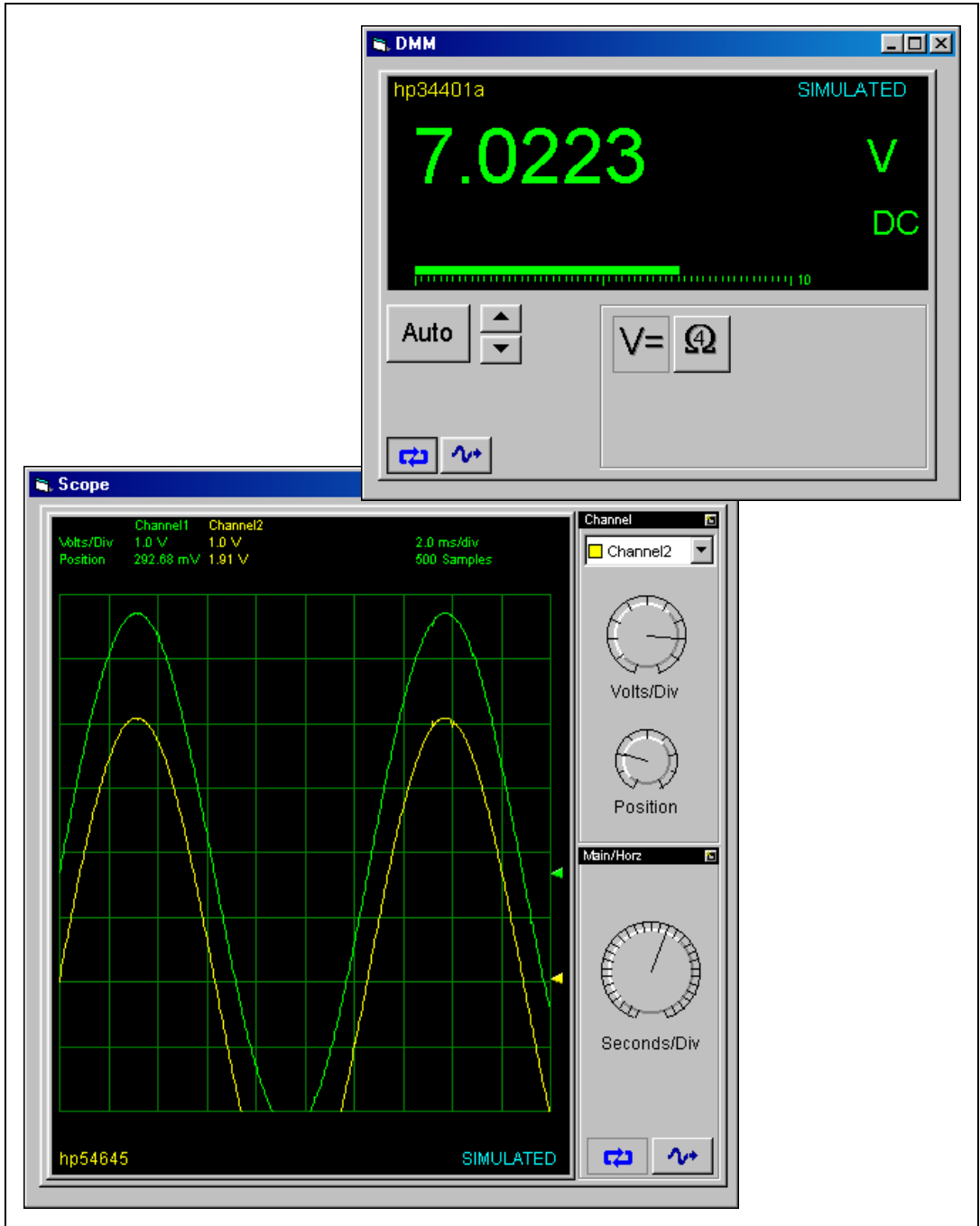


Figure 1-1. Measurement Studio Front Panels Represent Physical Instruments

# Measurement Studio

---

Measurement Studio offers LabWindows/CVI for ANSI C programmers, C++ classes and tools for Microsoft Visual C++ programmers, and ActiveX controls for Microsoft Visual Basic programmers.

## Measurement Studio LabWindows/CVI

LabWindows/CVI is a complete ANSI C development environment and compiler with built-in libraries for acquisition, analysis, and presentation. LabWindows/CVI provides a simplified drag-and-drop User Interface Editor and automated code generation tools that you can use to interactively test code before adding it to your project. With LabWindows/CVI, you can create localized user interfaces, work with ActiveX components, and create multithreaded applications. LabWindows/CVI delivers unique code generation utilities and the ability to create and control graphical user interfaces, control instruments and plug-in data acquisition hardware from your PC, and develop and debug ANSI C programs.

## Measurement Studio for Visual C++

Measurement Studio delivers an interactive design approach for developing virtual instruments in Visual C++. The Measurement Studio tools integrate into the Visual C++ environment so you can use them exactly as you would native Microsoft tools. Use the Measurement Studio AppWizard to create your measurement system. The AppWizard creates a project, according to your specifications, with an included code template and the measurement tools you need to design your application. These tools include C++ classes you use to interface with hardware, provide data analysis, and transfer data across the Internet, as well as custom COM-wrapped ActiveX controls you use to create your user interface. The link between the measurement classes and interface controls are data object classes that seamlessly encapsulate and pass data from acquisition to analysis to presentation.

## Measurement Studio for Visual Basic

Measurement Studio provides a collection of ActiveX controls designed for engineers and scientists building virtual instrumentation systems inside Visual Basic or other ActiveX control containers. With Measurement Studio, you can configure plug-in data acquisition boards, GPIB instruments, and serial devices from property pages without writing code. With user interface components, you can configure knobs, meters, gauges,

dials, tanks, thermometers, binary switches, LEDs, and 2D and 3D graphs. With powerful Internet components, you can share live measurement data between applications via the Internet.

## Related Software Packages

---

Measurement Studio offers additional packages for targeted applications. Visit the Product Catalog at [ni.com](http://ni.com) for more information about the following software packages:

### NI Developer Suite

For one package that includes all of the tools you need for developing an automated test application, consider NI Developer Suite. For example, the NI Developer Suite Professional Test Edition includes LabVIEW and Measurement Studio for developing test routines and TestStand for managing test execution, sequencing, collecting data, and generating reports. The test edition also includes a comprehensive set of LabVIEW and Measurement Studio add-on tools for Internet connectivity, database communication, signal processing, and code distribution.

### TestStand

TestStand is a ready-to-run test executive for organizing, controlling, and executing your automated prototype, validation, or manufacturing test systems. An off-the-shelf test executive, TestStand is completely customizable, so you can modify and enhance it to match your specific needs, including custom operator interface, report generation, and sequence execution requirements. Built on a high-speed, multithreaded, parallel execution engine, TestStand delivers high performance to meet your rigorous test throughput requirements.

### Vision and Image Processing Software

NI Vision and Image Processing Software includes IMAQ Vision, a library of vision functions, and IMAQ Vision Builder, an interactive environment for prototyping vision applications. Use vision and image processing software to build machine vision and scientific imaging applications.

## Motion Control Module

Develop motion control applications using your NI 733x or 734x series controller and a compatible ActiveX control container. This ActiveX control offers interactive property pages to help you quickly and easily configure moves. You can configure a single move or a sequence of moves and acquire position and velocity status for each move. You also can simultaneously start moves. As you configure your moves, the preview window changes to display trajectory and velocity. Pan, zoom, and rotate the preview window to inspect moves in three dimensions.

## IVI Driver Toolset

Instrument drivers, which are an important component in test systems today, perform the actual communication and control of the instrument hardware in your system. Instrument drivers provide a high-level, easy-to-use programming model. This programming model gives you complete access to the complex measurement capabilities of instruments via an intuitive API, thus delivering modular, off-the-shelf components ready to use in your test programs.



**Tip** Visit the Instrument Driver Network at [ni.com/idnet](http://ni.com/idnet) to find additional instrument drivers. Browse through instrument drivers by instrument type, manufacturer, or development language. Or, search for a specific instrument driver using the specifications most important to you—from a specific model name or manufacturer, to IVI drivers or drivers supported by NI.

## LabWindows/CVI Enterprise Connectivity Toolset

Enterprise connectivity tools help you track research and development, production, test, and servicing of products. The Enterprise Connectivity Toolset provides integrated tools for SQL database operations, SPC quality control, and Internet-enabling technology to give you the access and networking you need for business operations.

The following toolkits are included in the LabWindows/CVI Enterprise Connectivity Toolset:

- LabWindows/CVI Internet Developers Toolkit
- LabWindows/CVI SQL Toolkit
- LabWindows/CVI SPC Toolkit
- LabWindows/CVI Signal Processing Toolkit

## **Automation Symbols Toolset**

The Automation Symbols Toolset contains a collection of user interface ActiveX controls for industrial automation application interfaces—including valves, vessels, motors, pumps, and pipes. Each control has on/off states and events. You can configure the controls to use bitmaps as well as built-in animation.

## **PID Control Toolset**

The PID Control Toolset adds sophisticated control algorithms to your Measurement Studio applications. Implementing the standard PID algorithm, this toolset includes an autotuning wizard that assists you in optimizing the PID parameters to your specific system. Use the PID toolset with any kind of physical input and output options, including DAQ devices and PLCs.

# Getting Started with Measurement Studio

This chapter introduces the Measurement Studio Evaluation Package and provides system requirements and installation information.

## About this Evaluation Package

Evaluate Measurement Studio with your real-world test and measurement applications. The Measurement Studio Evaluation Package contains the Measurement Studio Full Development System with some restrictions, as listed in Table 2-1.

**Table 2-1.** Measurement Studio Evaluation Restrictions

Measurement Studio Product	Restrictions
LabWindows/CVI	<ul style="list-style-type: none"> <li>• 30-day expiration</li> <li>• 10-minute runtime timeout</li> <li>• No external compiler support</li> <li>• Disabled Create Distribution Kit feature</li> <li>• Disabled Create Object File feature</li> <li>• Disabled generation of import libraries from header files</li> <li>• No IVI Wizard support (disabled Create IVI Instrument Driver feature)</li> <li>• No Instrument Driver Only support</li> <li>• No LabVIEW Real-Time support</li> </ul>
Measurement Studio for Visual C++	<ul style="list-style-type: none"> <li>• 30-day expiration</li> <li>• 5-minute runtime timeout</li> </ul>
Measurement Studio for Visual Basic	<ul style="list-style-type: none"> <li>• Fully functional ActiveX controls for 30 days</li> <li>• 5-minute runtime timeout after 30 days</li> </ul>

The Full Development System includes the instrument control, user interface, advanced analysis, digital signal processing, and networking tools you need to start building test and measurement programs. With Measurement Studio, you can easily develop custom user interfaces to display your data; acquire data with your NI Data Acquisition (DAQ) boards; control serial, GPIB, and VXI instruments and controllers; analyze the data you acquired from a device; and share live data between different applications over the Internet.



**Note** To prevent the loss of your work, save all program files created with the Measurement Studio Evaluation Package and open them later with a licensed version.

## Minimum System Requirements

Table 2-2 lists the Measurement Studio minimum system requirements:

**Table 2-2.** Measurement Studio Minimum System Requirements

	Measurement Studio LabWindows/CVI	Measurement Studio for Visual C++	Measurement Studio for Visual Basic
Microprocessor	Pentium 133 MHz		
Operating Systems	Windows 2000/Me/9x Windows NT, Service Pack 3 or later		
Video Adapter	800 × 600	VGA resolution (or higher) video adapter; 16-bit color for User Interface and 3D Graph controls	
RAM	32 MB for Windows NT/9x 64 MB for Windows 2000/ME		
Free Disk Space*	175 MB	132 MB	85 MB
Third-Party Programming Environment	—	Microsoft Visual C++ 6.0, Service Pack 3 or later	32-bit ActiveX control container such as Microsoft Visual Basic
*Size required to install the Full Development System version of the Measurement Studio product. To install all three Measurement Studio products, you need 290 MB free disk space.			

# Installation Instructions

---

Unless you specify another location during installation, the Measurement Studio installation program copies files to `c:\Program Files\National Instruments\MeasurementStudio\` after you complete the following steps:

1. Insert the Measurement Studio CD in the CD drive of your computer. If the CD startup screen does not appear, use Windows Explorer to run `SETUP.EXE` from your CD.
2. Follow the instructions on your screen. On the Select Features screen, click any component that you do not want to install and select **Entire feature will be unavailable**.



**Note** If you decide later that you want to install additional features or components, run the installer again, click the feature or component you want to install, and select **Entire feature will be installed on local hard drive**.

3. Install driver software if you plan to use Measurement Studio with NI hardware.
4. Install your NI hardware. Refer to your hardware installation guide for installation information.
5. Configure your NI hardware with NI Measurement & Automation Explorer (MAX).

## Creating a New Project

---

Open your development environment—LabWindows/CVI, Visual C++, or Visual Basic—to start building a new virtual instrument. Use the following information to help you start developing your measurement application:

### LabWindows/CVI

To launch LabWindows/CVI, select **Start»Programs»National Instruments»Measurement Studio»LabWindows CVI**. When you open LabWindows/CVI, you see an empty Project window. The project file contains all of the files you need in your final compiled project. Like other Windows applications, use the **File»New** or **File»Open** commands to create new files or open existing files. To add files to the open project, use the **Edit»Add Files to Project** command.



As you develop applications in LabWindows/CVI, you work within three main windows: the Project window for \*.prj files, the Source window for \*.c files, and the User Interface Editor window for \*.uir files.

To complete a getting started tutorial for LabWindows/CVI, refer to the *Getting Started with Measurement Studio LabWindows/CVI* manual, which is available from **Start»Programs»National Instruments»Measurement Studio»Help»Measurement Studio Library** if you have LabWindows/CVI installed. Also spend some time examining installed samples, experimenting with LabWindows/CVI options and features, and reviewing related documentation listed in the Measurement Studio Library.

## Measurement Studio for Visual C++

Use the **File»New** command and choose the **Projects** tab as you would to create any other Visual C++ project. Select **NI Measurement Studio AppWizard**. The Measurement Studio AppWizard guides you through creating a project, allowing you to determine the design, the Measurement Studio for Visual C++ components to include, how to include the MFC libraries, and the MFC classes you want to use.

To complete a getting started tutorial for Measurement Studio for Visual C++, refer to the *Getting Started with Measurement Studio for Visual C++* manual, which is available from **Start»Programs»National Instruments»Measurement Studio»Help»Measurement Studio Library** if you have Measurement Studio for Visual C++ installed. Also spend some time examining installed samples, experimenting with the Measurement Studio wizards, and reviewing related documentation listed in the Measurement Studio Library.

## Measurement Studio for Visual Basic

Measurement Studio for Visual Basic offers four predefined project templates that you use to create a new measurement application when you open Visual Basic or select **File»New**. Each project template automatically loads a defined set of Measurement Studio controls. Refer to Table 2-3 for information about each Measurement Studio project template.

**Table 2-3.** Measurement Studio for Visual Basic Project Templates

Template	Components	Use
NI Instrumentation EXE	3D Graph, Analysis, DataSocket, Instrumentation, IVI, User Interface, and VISA	Stand-alone instrumentation applications
NI Instrumentation ActiveX Control	3D Graph, Analysis, DataSocket, Instrumentation, IVI, User Interface, and VISA	Web instrumentation applications
NI Measurements EXE	3D Graph, Analysis, DataSocket, Data Acquisition, and User Interface	Stand-alone measurements applications
NI Measurements ActiveX Control	3D Graph, Analysis, DataSocket, Data Acquisition, and User Interface	Web measurements applications

To complete a getting started tutorial for Measurement Studio for Visual Basic, refer to the *Getting Started with Measurement Studio for Visual Basic* manual, which is available from **Start»Programs»National Instruments»Measurement Studio»Help»Measurement Studio Library** if you have Measurement Studio for Visual Basic installed. Also spend some time examining installed samples, experimenting with the Measurement Studio property pages, and reviewing related documentation listed in the Measurement Studio Library.

---

# Exploring Measurement Studio

This chapter explores some of the tools that Measurement Studio provides for rapid virtual instrument development. Each section contains an exercise to guide you. After you complete all exercises for a given programming environment, you have a simple program that plots data on a graph.

Although this example only begins to reveal the power of Measurement Studio, it is a good place to start learning about the Measurement Studio tools.

---

## Interactive C Programming Environment

---

If you are already programming with C, LabWindows/CVI complements your existing efforts and streamlines your future development. Because LabWindows/CVI is built on an open software architecture, you can reuse existing programs within the LabWindows/CVI environment, and you can incorporate standard ANSI C code, object files, and DLLs in your application.



**You can complete this activity in 5 minutes.**

1. Launch LabWindows/CVI. When you open LabWindows/CVI, you see an empty Project window. The project file contains all of the files you need in your final compiled project.
2. Select **File»New»User Interface** to create a new UIR file, and save it.
3. Add the new UIR file to the empty project.
4. Explore the three main windows in LabWindows/CVI.

For information about using the LabWindows/CVI environment, refer to *Using LabWindows/CVI* from the Contents of the LabWindows/CVI Help.

## Microsoft Visual Studio Integration

---

NI is committed to making your programming tasks easier using today's popular development environments. With this goal in mind, NI integrates measurement tools into Visual Studio—specifically Microsoft Visual Basic and Visual C++—so you can benefit from measurement expertise without learning a new language. To integrate more efficiently into the Visual Studio development environment, NI formed an alliance with Microsoft through the Visual Studio Integration Program (VSIP). Through this program, NI acquires the tools necessary to achieve more seamless integration with Microsoft environments. As a developer, you have less to learn and can start using our tools quickly.

### Measurement Studio Wizards in Visual C++

Measurement Studio provides integrated wizards to help you create new projects complete with measurement tools you need, modify existing projects to include the measurement tools you need, and much more.



**You can complete this activity in 5 minutes.**

1. Launch Visual C++, and select **File»New»Projects**. Notice that you can select a new type of application—**NI Measurement Studio AppWizard**.
2. Select it, and create a new project with the Measurement Studio components you need to build your application.
3. View the Controls toolbar in the dialog resource editor and identify the Measurement Studio User Interface controls. All Measurement Studio user interface control names start with **CW**. For example, look for CWButton Control.
4. View the MStudio toolbar in the Visual C++ application window to take a look at the other Measurement Studio wizards. For information about Measurement Studio wizards, refer to *Wizards* from the Index of the Measurement Studio Reference.

### Measurement Studio Project Templates in Visual Basic

Measurement Studio offers four predefined project templates that you can use to create a new Visual Basic project. Each project template automatically loads a defined set of Measurement Studio controls for you, so you can start developing an executable or ActiveX control as soon as your new project opens.



**You can complete this activity in 5 minutes.**

1. Launch Visual Basic and select one of the Measurement Studio EXE templates. Although both templates contain User Interface controls, the Measurements template includes ActiveX controls for data acquisition, and the Instrumentation template includes ActiveX controls for GPIB, serial, VISA, and IVI interfaces.
2. View the Toolbox, and identify the Measurement Studio ActiveX controls. The control names start with **CW**. For example, look for CWButton. For more information about Visual Basic templates, refer to *Project Templates* from the Index of the Measurement Studio Reference.

## Measurement Libraries and Components

---

Measurement Studio delivers I/O, analysis, presentation, and other measurement-specific libraries and components to help you build your test, measurement, or control system through intuitive interfaces—function panel libraries for LabWindows/CVI, MFC-based C++ classes for Visual C++, or ActiveX interfaces for Visual Basic.

For a complete list of all libraries and components, refer to the Quick Reference Cards included with this evaluation package.

## Data Acquisition and Instrument Control

Whatever you use to acquire your data—a GPIB or serial instrument, a plug-in data acquisition card, PXI hardware, NI computer-based instruments, or an image acquisition device—Measurement Studio provides the high-level interface native to your development environment.

## Data Analysis

Measurement Studio delivers the analysis functions you need to convert raw data into meaningful information. With Measurement Studio, you can employ a wide variety of signal processing and data analysis tools, such as curve fitting, spectral analysis, statistics, and visualization, for research and development, engineering and validation, or manufacturing and service applications. Because you can perform your analysis inside your application, you do not need another application for post-process analysis.

## Presentation

Measurement Studio delivers ready-to-use, measurement-specific user interface components with which you can quickly and easily create professional measurement applications that mimic stand-alone instruments, such as scopes and temperature loggers. Use Measurement Studio 3D graphics to create truly revolutionary application interfaces.

## Networking

Whether you pass data between applications or over the Internet, Measurement Studio makes sharing measurement data easy. With NI DataSocket, a programming tool designed for publishing and subscribing to live data in measurement and automation applications, you can publish or retrieve data, control remote applications, or distribute your system with one or more client applications on a network without worrying about data formats and network protocols.

Connect to data sources via the following interfaces from applications you develop with Measurement Studio:

- DSTP (DataSocket server transfer protocol)
- OPC
- HTTP
- FTP
- File



**Tip** The Measurement Studio User Interface ActiveX controls include built-in DataSocket connectivity. You can create a simple yet powerful Visual Basic or Visual C++ program to visualize measurements or publish them from your desk, and you can do it without writing code. For information about built-in DataSocket connectivity, browse through the Bindings property page of a Measurement Studio User Interface control in Visual C++ or Visual Basic.

## Explore Measurement Libraries and Components



You can complete these activities in 5 minutes.

### LabWindows/CVI

1. To survey the libraries offered in LabWindows/CVI, view the **Library** menu items.
2. Select one of the libraries, and scroll through the functions that library has to offer.
3. Open one of the functions and examine the function panel. The function panel graphically displays the parameters for this function.



**Tip** Right click a parameter or anywhere in the panel for help about a specific parameter or the function.

### Measurement Studio for Visual C++ and Visual Basic

Use the Measurement Studio Reference to view a list of all measurement components available to you in Visual C++ and Visual Basic.

1. Launch the Measurement Studio Reference from **Start»Programs»National Instruments»Measurement Studio»Help**.
2. Scroll through the **Visual Basic Help** and **Visual C++ Help Reference** folders in the Contents tab.
3. Select any of the objects listed in the Contents and review the properties and methods you can use to programmatically control that object.

## Configurable User Interface Controls

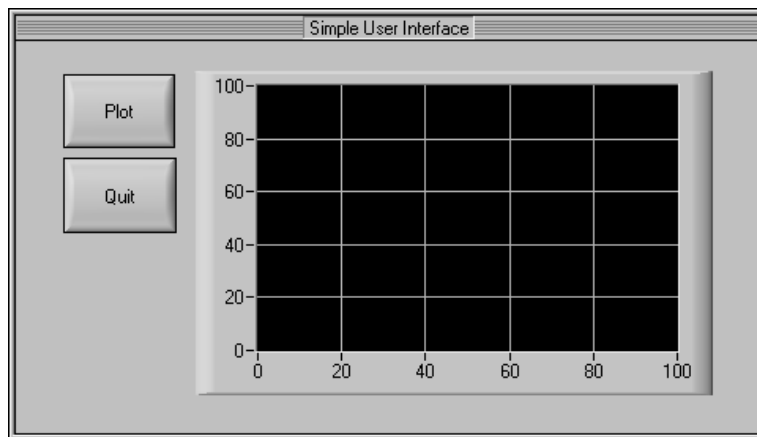
---

Measurement Studio provides a collection of highly configurable, instrument-style user interface controls and indicators so users of your application can interact with and monitor the measurement or control system. Choose from the following controls to create your front panels:

- knobs, meters, gauges, dials
- binary switches, LEDs
- slides, tanks, thermometers
- real-time 2D and 3D graphs
- tables (LabWindows/CVI only)

## Create a Simple UI with a Graph and Buttons

In this activity, you create a simple user interface that contains a graph and two command buttons. One button plots a waveform on the graph, and the second button stops the application. Your user interface might look like the one shown in Figure 3-1, which was created in LabWindows/CVI.



**Figure 3-1.** Simple User Interface Created in LabWindows/CVI



**Note** Name all three controls according to conventions in your programming language. For example, in Visual C++, you might name the graph `m_graph`, the **Plot** button `m_plot`, the **Quit** button `m_quit` according to member variable naming conventions. In LabWindows/CVI, you might name the controls `GRAPH`, `PLOT`, and `QUIT`, respectively. In Visual Basic, you might name keep the default name for the graph, `CWGraph1`, and name the pushbuttons `btnPlot` and `btnQuit`.



**You can complete the activity for one environment in 15 minutes.**

### LabWindows/CVI

1. In the User Interface Editor, right click the panel and choose one of the graph styles from the menu. LabWindows/CVI places the graph on the panel.
2. Double click the graph to modify its properties. Name the control and then experiment with the axes, grid, labels, and application startup properties.



3. Add a command button to the user interface. Name and configure it as the Plot button.
4. Add a Quit button from Custom Controls by right clicking the panel and selecting **Custom Controls»Quit Button**.

## Visual C++

1. Drag and drop a Measurement Studio Graph (CWGraph Control) from the Controls toolbar to the resource dialog. Right click the graph, and select **CWGraph Control (National Instruments) Object»Properties**. Inspect the property pages and experiment with the properties. After you configure the graph, close the property pages.



**Tip** Use the preview window to see how different property values affect the control before you apply them.

2. Drag and drop a Measurement Studio Button on the dialog and configure the Plot button.
3. Delete the OK button but leave the Cancel button. Use the Cancel button as the Quit button.

## Visual Basic

1. Place a Measurement Studio Graph (CWGraph) from the Toolbox on the form. Right click the graph, and select **Properties**. Inspect the property pages and experiment with the properties. After you configure the graph, close the property pages.



**Tip** Use the preview window to see how different property values affect the control before you apply them.

2. Place a Measurement Studio Button on the dialog and configure the Plot button.
3. Add the Quit button.

## Context-Sensitive Help

---

NI designed Measurement Studio to deliver the appropriate information to you when you need it. In addition to traditional help files, you will find context-sensitive help throughout Measurement Studio. Look for and take advantage of the help Measurement Studio provides, including **Help** buttons, F1 connectivity, or **What's This?** help in Measurement Studio

wizards, dialog boxes, menus, property pages, and source windows. The following sections provide more information about context-sensitive help available in each environment.



**Note** You can launch the LabWindows/CVI Help and the Measurement Studio Reference from **Start»Programs»National Instruments»Measurement Studio»Help**.

## LabWindows/CVI

- **Help file**—Select **Help»Contents** in LabWindows/CVI to access the help file.
- **Menus, dialog boxes, and Source window keywords**—Press <F1> on menus, dialog boxes, or function names in the Source window. You also can right click menu items and select **Menu Help** to learn more about the options and features available in that menu.
- **Function panels**—Right click the panel or any parameter for help about that function or parameter.

## Measurement Studio for Visual C++

- **Help file**—Select **Tools»Measurement Studio Reference** in Visual C++ to access the Measurement Studio Reference help file.
- **User Interface control help**—Right click Measurement Studio controls in a dialog resource and select **ControlName Object»Help**, where *ControlName* is the name of the control that you clicked.
- **Property pages**—Right click properties in the property pages and select **What's This?** to learn more about the property. Several properties also include code examples that demonstrate how to control the property programmatically.
- **Source code keywords**—In the Microsoft Visual C++ source code editor, highlight or place your cursor in a Measurement Studio class, function, or property name. Select **Tools»Measurement Studio Reference** to view information about the selected keyword.

## Measurement Studio for Visual Basic

- **User Interface control help**—Right click Measurement Studio controls on a form and select **Help**. Alternatively, you can right click the control, select **Properties**, and then click the **Help** button.
- **Property pages**—Right click properties in the property pages and select **What's This?** to learn more about the property. Several

properties also include code examples that demonstrate how to control the property programmatically.

- **Source files**—Highlight or place your cursor in any Measurement Studio object, property, method, or event name and press <F1> to open the function reference for that keyword.

## LabWindows/CVI Code Generation

---

LabWindows/CVI offers a unique code generation tool called CodeBuilder. CodeBuilder uses information that you specify in control and event dialog boxes to create a code skeleton for you. The code skeleton includes the `main` function, event callbacks, and an application shutdown callback.

LabWindows/CVI provides event-driven programming through callback functions. You can associate a specific callback function with a specific user interface control in the control's Edit dialog box. As you set the properties for the control, think about what events you want the control to recognize—such as a single mouse click, double click, or keypress—and specify the callback function that executes when those events occur.

Specify default events for both the panel and controls on the panel. You can enable events using the **Code»Preferences»Default Panel Events** and **Default Control Events**. When you generate your code, CodeBuilder creates a case statement for each event you enable—you just fill in the code to handle that event.



**You can complete this activity in 20 minutes.**

1. Open the Edit Command Button dialog boxes for the Plot command button and add a callback function name.
2. Specify your default control events. For this simple example, you will handle only a single mouse click event.
3. Select **Code»Generate»All Code** to have LabWindows/CVI generate the code for you. Be sure to indicate the callback function that terminates the program when you press the **Quit** button.



**Tip** Click the **Help** button in the dialog boxes to learn more about default events and generating code. For more information about events, refer to *Events Overview* from the Index of the LabWindows/CVI Help.

## Simplified APIs and Native Data Types

Measurement Studio offers intuitive, language-specific interfaces to its libraries and components. LabWindows/CVI uses standard C data types and function panels. Measurement Studio for Visual C++ provides C++ classes that use C++ data types and some helper classes to help you deal with data types such as Variants and SAFEARRAYS when you have to use them. Measurement Studio for Visual Basic provides ActiveX controls with properties, methods, and events that take full advantage of Variants. You do not have to worry about specifically typing or converting data types when passing variables to methods or using returned values in events.



**You can complete this activity in 20 minutes.**

Add the following functionality to complete your simple data plotting example:

1. Acquire or simulate a waveform to plot on the graph. You can acquire data from NI hardware or from across the network using DataSocket. If you need to simulate a waveform, consider using a sine wave function from the Measurement Studio analysis library.



**Tip** If you are working in LabWindows/CVI, take advantage of function panels. Open the Sine Wave function panel from the Project window (**Library»Advanced Analysis»Signal Generation»Sine Wave**), fill in the parameters, and select **Code»Insert Function Call** to add the call to your code. For information about working with function panels, refer to *Using Function Panels* from the Index of the LabWindows/CVI Help.

2. Plot the data. Measurement Studio Graph controls offer several different functions for plotting and charting data. Use the LabWindows/CVI function panels and the Measurement Studio Reference to learn more about these functions and then choose the one that works best for the application.
3. Properly shut down the application.
  - LabWindows/CVI—CodeBuilder created a callback that calls `QuitUserInterface` when you click the **Quit** button. You do not need to implement anything for this step.
  - Visual C++—The MFC framework automatically provides the Quit functionality with the Cancel button. You do not need to implement anything for this step.
  - Visual Basic—Create a `Click` event procedure for the **Quit** button and call the Visual Basic `End` function.

# Distribution

---

When you need to distribute applications built with Measurement Studio, take advantage of the LabWindows/CVI Create Distribution Kit feature, the merge modules and run-time files for MSI-based installer applications, and the Visual Studio Package & Deployment Wizard.

The Create Distribution Kit feature makes an installer application from which you can install your LabWindows/CVI executable program on a target machine. Create Distribution Kit automatically includes most files necessary to run your executable program on a target computer. For more information about distributing applications you build with LabWindows/CVI, refer to *Create Distribution Kit* from the Index of the LabWindows/CVI Help.



**Note** The LabWindows/CVI Create Distribution Kit feature is disabled in the Measurement Studio Evaluation Package.

For information about distributing applications developed with Measurement Studio for Visual C++ or Measurement Studio for Visual Basic, refer to *Distribution* and *Redistribution* from the Index of the Measurement Studio Reference.

---

## Where to Go from Here

Measurement Studio is yet another long-standing commitment by NI to provide tools that simplify the development of instrumentation, data acquisition, and control systems. When you choose Measurement Studio, you join thousands of scientists and engineers who take advantage of the power of these versatile tools for quick and easy test and measurement development in the language of their choice.

Visit the Measurement Studio Web site and NI Developer Zone at [ni.com/zone](http://ni.com/zone) for the most up-to-date information.

---

## Tutorials and Related Documentation

Read the Getting Started manual for your Measurement Studio product. The Measurement Studio Evaluation CD contains PDF files of the following manuals:

- *Getting Started with Measurement Studio LabWindows/CVI*
- *Getting Started with Measurement Studio for Visual C++*
- *Getting Started with Measurement Studio for Visual Basic*



**Note** Use Adobe Acrobat Reader 3.0 or later to view any Measurement Studio PDF file. You can download a free version of the Reader from [www.adobe.com](http://www.adobe.com) or install a copy from your Measurement Studio CD.

Visit the Measurement Studio Library, available from **Start»Programs»National Instruments»Measurement Studio»Help**. The Library lists all currently available Measurement Studio documentation organized by development environment. The Library also describes and links to Measurement Studio manuals, application notes, and white papers in PDF format.

## Programmer and Reference Information

---

The LabWindows/CVI Help and Measurement Studio Reference for Visual C++ and Visual Basic contain complete reference information.

LabWindows/CVI Help contains the following sections:

- *Using LabWindows/CVI* describes each window and dialog box in the LabWindows/CVI.
- *Library Reference* includes reference for functions you can call in LabWindows/CVI.
- *Programmer Reference* provides information to help you develop applications.
- *Example Programs* contains LabWindows/CVI samples you can launch and run from the help topic.
- *Tools Library* lists instrument drivers included with LabWindows/CVI.

Measurement Studio Reference contains the following sections:

- *Visual Basic Help* includes reference information to help you programmatically control the Measurement Studio ActiveX controls in Visual Basic.
- *Visual C++ Help* includes the Measurement Studio for Visual C++ API reference.
- *Tutorials* provides a combination of application notes, getting started materials, and procedures for using Measurement Studio in Visual Basic and Visual C++.

You can launch either help file from **Start»Programs»National Instruments»Measurement Studio»Help**.

Also, review the Quick Reference Cards provided with this evaluation package. They list the LabWindows/CVI libraries and Measurement Studio C++ classes and ActiveX controls available to you.

## Measurement Studio Examples

---

Measurement Studio installs example programs for each development environment. You also can visit NI Developer Zone at [ni.com/zone](http://ni.com/zone) for new examples.



**Tip** Examples are an excellent way to get started. Look for an example similar to what you need and modify it according to your specifications.

## Customer Education

---

For additional training, NI offers hands-on LabWindows/CVI courses to help you quickly master the environment and develop successful applications.





---

# Technical Support Resources

## Web Support

---

National Instruments Web support is your first stop for help in solving installation, configuration, and application problems and questions. Online problem-solving and diagnostic resources include frequently asked questions, knowledge bases, product-specific troubleshooting wizards, manuals, drivers, software updates, and more. Web support is available through the Technical Support section of [ni.com](http://ni.com).

## NI Developer Zone

---

The NI Developer Zone at [ni.com/zone](http://ni.com/zone) is the essential resource for building measurement and automation systems. At the NI Developer Zone, you can easily access the latest example programs, system configurators, tutorials, technical news, as well as a community of developers ready to share their own techniques.

## Customer Education

---

National Instruments provides a number of alternatives to satisfy your training needs, from self-paced tutorials, videos, and interactive CDs to instructor-led hands-on courses at locations around the world. Visit the Customer Education section of [ni.com](http://ni.com) for online course schedules, syllabi, training centers, and class registration.

## System Integration

---

If you have time constraints, limited in-house technical resources, or other dilemmas, you may prefer to employ consulting or system integration services. You can rely on the expertise available through our worldwide network of Alliance Program members. To find out more about our Alliance system integration solutions, visit the System Integration section of [ni.com](http://ni.com).

## Worldwide Support

---

National Instruments has offices located around the world to help address your support needs. You can access our branch office Web sites from the Worldwide Offices section of [ni.com](http://ni.com). Branch office Web sites provide up-to-date contact information, support phone numbers, e-mail addresses, and current events.

If you have searched the technical support resources on our Web site and still cannot find the answers you need, contact your local office or National Instruments corporate. Phone numbers for our worldwide offices are listed at the front of this manual.

# Glossary

---

## A

ActiveX	Set of Microsoft technologies for reusable software components. Formerly called <i>OLE</i> .
ActiveX control	Reusable software component that adds functionality to any ActiveX control container through exposed properties, methods, and events. The Measurement Studio Data Acquisition and User Interface controls are examples of ActiveX controls.
ActiveX control container	Development environment that fully supports ActiveX controls and integrates them into its own environment using COM. An ActiveX control container enables you to specify how ActiveX controls interact with the environment through environment properties. Visual Basic is an example of an ActiveX control container.
ANSI C	C programming language defined by the American National Standards Institute.
API	Application Programming Interface. A specification of software functions and their input and return parameters.

## C

callback	User-defined function called in response to an event from a user interface control or object. Also called an <i>event handler</i> in Visual C++ or <i>event procedure</i> in Visual Basic. <i>See also</i> <a href="#">event</a> .
CodeBuilder	LabWindows/CVI feature that creates code based on a <code>.uir</code> file to connect your GUI to the rest of your program. This code can be compiled and run as soon as it is created.
COM	Component Object Model. Microsoft technology that allows developers to build applications from binary software components. COM is the underlying architecture that forms the foundation for higher-level software services such as OLE.
ComponentWorks	Former name of Measurement Studio for Visual Basic. <i>See</i> <a href="#">Measurement Studio</a> .

context-sensitive help	Help for dialog boxes, the controls in dialog boxes, and keywords in source code that you can access with the <F1> key or a <b>Help</b> button, by right clicking a control and selecting <b>What's This?</b> , or by clicking the ? (question mark) in the top right of a dialog box and then clicking on a control with which you need help.
control	<ol style="list-style-type: none"><li>1. Object for entering or manipulating data on a user interface. <i>Compare with <a href="#">indicator</a>.</i></li><li>2. <i>See <a href="#">ActiveX control</a>.</i></li></ol>
control property	Property that defines the way an ActiveX control looks and behaves. You can set control properties programmatically or with custom property pages.
cursor	Flashing rectangle that shows where you may enter text on the screen. If you have a mouse installed, there is a rectangular mouse cursor, or pointer.
<b>D</b>	
DAQ	Data acquisition. Process of acquiring data, typically from A/D or digital input plug-in boards.
DataSocket	National Instruments technology that simplifies live data exchange between applications and DataSocket, HTTP, FTP, OPC, and file servers through one common API.
distribution	Ability to install programs you develop with Measurement Studio to others working on different computers.
DLL	Dynamic Link Library. A library of functions that link to a program and load at runtime rather than being compiled into the program. Loading libraries only when they are needed saves memory in software applications.
DMM	Digital Multimeter. A common measurement instrument that measures resistance, current, and voltage in a wide variety of applications.
driver	<i>See <a href="#">instrument driver</a>.</i>

**E**

Ethernet	Standard connection type for networks, where computers are connected by coaxial or twisted-pair cable.
event	Object-generated response to some action or change in state, such as a mouse click or a completed acquisition. The event calls an event procedure, handler, or callback that processes the event.
event-driven	Describes a program that runs continuously, waiting for an event to occur. When an event occurs, the program calls the appropriate event procedure. Compare with <a href="#">loop-driven</a> , where a program continuously polls to find out if anything has changed.
event handler	<i>See</i> <a href="#">callback</a> .
event procedure	<i>See</i> <a href="#">callback</a> .

**F**

form	Window or area on the screen on which you place controls and indicators to create the user interface for your program.
front panel	Interactive user interface of a virtual instrument. Modeled after the front panel of physical instruments, it is composed of switches, slides, meters, graphs, charts, gauges, LEDs, and other controls and indicators.
function panel	A screen-oriented user interface to the LabWindows/CVI libraries that allows interactive execution of library functions and is capable of generating code for inclusion in a program.

**G**

GPIB	General Purpose Interface Bus. The standard bus used for controlling electronic instruments with a computer. Also called <i>IEEE 488 bus</i> because it is defined by ANSI/IEEE Standards 488-1978, 488.1-1987, and 488.2-1987.
GUI	Graphical User Interface. A user interface that has graphics, controls, indicators, and/or menus.

## I

IMAQ Vision	National Instruments image acquisition and analysis software that you can use to acquire images from National Instruments image acquisition (IMAQ) boards, display them in your program, perform interactive viewer operations, and analyze the images to extract information.
indicator	Object for displaying data on a user interface. <i>Compare with</i> <a href="#">control</a> .
installer	Software program that copies program, system, and other necessary files to computers. You can use tools such as the Microsoft Visual Studio Package & Deployment Wizard to create installers for programs you want to distribute to others.
instrument driver	Library of functions to control and use one specific physical instrument. Also a set of functions that adds specific functionality to an application.
interface	<ol style="list-style-type: none"><li>1. Connection between one or more of the following: hardware, software, and the user. For example, hardware interfaces connect two other pieces of hardware.</li><li>2. <i>See</i> <a href="#">GUI</a>.</li></ol>
IVI	Interchangeable Virtual Instruments. A technology involving standard programming interfaces for classes of instruments, such as oscilloscopes, DMMs, and function generators, that results in hardware-independent instrument drivers. The IVI standard programming interfaces have been defined by the IVI Foundation, an industry consortium. Visit <a href="http://www.ivifoundation.org">www.ivifoundation.org</a> .

## L

LabVIEW	Laboratory Virtual Instrument Engineering Workbench. Program development application based on the programming language G and used commonly for test and measurement purposes.
LabWindows/CVI	Measurement Studio product for ANSI C developers who build test and measurement applications.
LED	Light-Emitting Diode. An indicator that emits a light when current passes through it. For example, an LED shows if your computer or printer is turned on.

loop-driven Describes a program that uses a main loop to keep polling the program for changes. *Compare with* [event-driven](#).

## M

MAX Measurement & Automation Explorer. National Instruments tool for configuring your National Instruments hardware and driver software; executing system diagnostics; adding new devices, interfaces, and virtual channels; and viewing devices and instruments connected to your system.

Measurement Studio National Instruments software that includes tools to build measurement applications in LabWindows/CVI, Visual C++, and Visual Basic.

method Function that performs a specific action on or with an object. The operation of the method often depends on the values of the object properties.

MFC Microsoft Foundation Class. A framework for programming in Microsoft Windows, MFC provides code for managing windows, menus, and dialog boxes; performing basic input/output; storing collections of data objects; and more.

## N

NI-488.2 Driver-level software to control and communicate with National Instruments GPIB hardware.

NI-DAQ Driver-level software to control and communicate with National Instruments data acquisition hardware.

NI-IMAQ Driver-level software to control and communicate with National Instruments image acquisition hardware.

## O

- OPC OLE for Process Control. An industry standard based on ActiveX and COM technologies that enables you to create a single client application that can communicate with disparate devices. Visit [www.opcfoundation.org](http://www.opcfoundation.org).
- oscilloscope Measurement instrument widely used in high-speed testing applications, such as telecommunication physical layer testing, video testing, and high-speed digital design verification.

## P

- PCI Peripheral Component Interconnect. High-performance expansion bus architecture commonly found in PCs.
- PID Proportional-Integral-Derivative. A three-term control mechanism combining proportional, integral, and derivative control. You might use a PID algorithm to control processes such as heating and cooling systems, fluid level monitoring, flow control, and pressure control.
- preview window Measurement Studio tool for experimenting with User Interface ActiveX control properties in the property pages before applying them to the application.
- property Attribute that defines the appearance or state of an object.
- property pages Window or dialog box that displays current configuration information and allows users to modify the configuration. Also called *property sheets*.
- PXI PCI eXtensions for Instrumentation. Rugged, open system for modular instrumentation with specialized mechanical, electrical, and software features.

## S

- scope *See* oscilloscope.
- serial Standard serial bus on a computer used to communicate with instruments. Also known as *RS-232*.



**T**

**TestStand** Ready-to-run test executive from National Instruments for organizing, controlling, and executing your automated prototype, validation, or manufacturing test systems.

**U**

**UI** User Interface. *See* [GUI](#).

**V**

**virtual instrument** Combination of hardware and/or software elements, typically used with a PC, that has the functionality of a classic stand-alone instrument.

**VISA** Driver-software architecture developed by National Instruments to unify instrumentation software for serial, GPIB, and VXI instruments or controllers. It has been accepted as a standard for VXI by the *VXIplug&play* Systems Alliance.

**VXI** VME eXtension for Instrumentation. Instrumentation architecture and bus based on the VME standard. Used in high-end test applications.