Measurement Studio™

User Manual

June 2003 Edition
Part Number 323392A-01

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 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599,
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, Canada (Vancouver) 514 685 7530,
China 86 21 6555 7838, Czech Republic 420 2 2423 5774, Denmark 45 45 76 26 00,
Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, Greece 30 2 10 42 96 427,
India 91 80 51190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970,
Korea 82 02 3451 3400, Malaysia 603 9131 0918, Mexico 001 800 010 0793, Netherlands 31 0 348 433 466,
New Zealand 1800 300 800, Norway 47 0 66 90 76 60, Poland 48 0 22 3390 150, Portugal 351 210 311 210,
Russia 7 095 238 7139, Singapore 65 6226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197,
Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 2 2528 7227,
Thailand 662 992 7519, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on the documentation, send email to techpubs@ni.com.

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 THERETOFORE 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

Trademarks

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

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.

Contents

About This Manual

The *Measurement Studio User Manual* introduces the concepts associated with the Measurement Studio class libraries and development tools. This manual assumes that you have a general working knowledge of Microsoft Visual Studio .NET, including Microsoft Visual Basic .NET, Microsoft Visual C#, or Microsoft Visual C++ .NET.

How to Use this Manual

The *Measurement Studio User Manual* is organized into four chapters. Chapter 1, *Introduction to Measurement Studio*, is an overview of Measurement Studio, which includes installation and distribution requirements, installation instructions, and a list of Measurement Studio resources. Chapter 2, *Measurement Studio .NET Class Libraries*, and Chapter 3, *Measurement Studio Visual C++ Class Libraries*, include information about the .NET class libraries and the Visual C++ class libraries, respectively. Chapter 4, *Developing with Measurement Studio*, includes information on developing applications with Measurement Studio tools and features.

Use this manual as a starting point to learn about Measurement Studio. Refer to the *NI Measurement Studio Help* within the Visual Studio .NET environment for function reference and detailed information about the Measurement Studio class libraries, wizards, assistants, and other features.

Conventions

The following conventions appear 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 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 in the software, such as menu items and dialog box options.

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 enter from the keyboard, sections of code, programming examples, and syntax examples. This font also is used for the proper names of disk drives, paths, directories, programs, device names, filenames and extensions, and code excerpts.

1

Introduction to Measurement Studio

Measurement Studio is an integrated suite of native tools and class libraries that are designed for developers using Visual Basic .NET, Visual C#, and Visual C++ .NET to develop measurement and automation applications.

Measurement Studio dramatically reduces application development time through object-oriented measurement hardware interfaces, advanced analysis libraries, scientific user interface controls, measurement data networking, wizards, interactive code designers, and highly extensible .NET and Visual C++ classes. You can use Measurement Studio to develop a complete measurement and automation application that includes data acquisition, analysis, and display functionalities.

Installation Requirements

To install Measurement Studio, your computer must have the following:

- Microsoft Windows 2000/XP operating system (Windows 2000 users must have the latest Windows service pack and critical updates available from the Microsoft Security Web site.)
- Microsoft .NET Framework 1.1 (required only for the Measurement Studio .NET class libraries)
- Professional, Enterprise Developer, or Enterprise Architect edition of Microsoft Visual Studio .NET 2003 (required to use the Measurement Studio integrated tools)
- Appropriate hardware if you are going to use a Measurement Studio I/O hardware class library
- Intel Pentium class processor, 133 MHz or higher
- Video display—800 × 600, 256 colors (16-bit color recommended for user interface controls)
- Minimum of 128 MB of RAM (256 MB or higher recommended)
- Minimum of 200 MB of free hard disk space

- Microsoft-compatible mouse
- Microsoft Internet Explorer 5.01 or later

Optional Installation—In order for links from Measurement Studio help topics to .NET Framework help topics to work, you must install the Microsoft .NET Framework SDK 1.1.

Distribution Requirements

To distribute an application built with Measurement Studio .NET class libraries, the target computer must have a Windows 2000/XP operating system and .NET Framework 1.1.

To distribute an application built with Measurement Studio Visual C++ class libraries, the target computer must have a Windows 2000/XP operating system.

Installation Instructions

Complete the following steps to install Measurement Studio.

1. Insert the Measurement Studio CD into the CD-ROM. `autorun.exe` automatically starts. If it does not automatically start, double-click the `autorun.exe` icon.
2. Click **Install NI Measurement Studio**.
3. Enter the serial number and click **Next**.
4. Review the license agreement and select **I accept the License Agreement(s)**. Click **Next**.
5. Click **Next** to install all NI software to the default installation directory, or click **Browse** to select a different installation directory. You must install Measurement Studio to a local drive. Click **Next**.

Note The option to browse for an installation location is valid only if you have not already installed any Measurement Studio features of the same version you are installing. If you have any Measurement Studio features of the same version installed, the Measurement Studio installer installs to the same root directory to which you installed other Measurement Studio features.

6. From the feature tree, select the components you want to install. To change the Measurement Studio 7.0 installation directory, select the first feature in the list and click **Browse**. Click **Next**.

Note Measurement Studio supports side-by-side installation. If you would like to install different versions of Measurement Studio, you must choose a different directory for each version of Measurement Studio.

7. In the Installation Summary dialog box, review the features you selected. Click **Next**.
8. If prompted, insert the Device Drivers CD and select **Rescan Drive**. If not prompted, go to step 11 on this list.
9. From the feature tree, select the Device Drivers components you want to install. To change a driver installation directory, select the driver and click **Browse**. Click **Next**.
10. In the Installation Summary dialog box, review the features you selected. Click **Next**.
11. When prompted, click the appropriate registration options.
12. Click **Finish** to complete the installation.
13. Click the appropriate restart option.

Measurement Studio Resources

As you work with Measurement Studio, you might need to consult other resources. For detailed Measurement Studio help, including function reference and in-depth documentation on developing with Measurement Studio, refer to the *NI Measurement Studio Help* within the Visual Studio .NET environment. The *NI Measurement Studio Help* is fully integrated with the Visual Studio .NET help. You must have Visual Studio .NET installed to view the online help. You can launch the *NI Measurement Studio Help* in the following ways:

- From the Windows Start menu, select **Start»Programs»National Instruments»Measurement Studio 7.0»Measurement Studio Documentation**. The help launches in a stand-alone help viewer.
- From Visual Studio .NET, select **Help»Contents** to view the Visual Studio .NET table of contents. The *NI Measurement Studio Help* is listed in the table of contents.

Tip As you work through this manual, you will see italicized references to relevant help topics. You can find these topics by using the table of contents in the *NI Measurement Studio Help*.

The following sources also are available to provide you with information about Measurement Studio.

- Examples—Measurement Studio installs examples to the following paths:
  - Visual Basic .NET or Visual C#—`Program Files\National Instruments\MeasurementStudio70\DotNET\Examples`
  - Visual C++ .NET—`Program Files\National Instruments\MeasurementStudio70\VCNET\Examples`
- NI Technical Support—Refer to Appendix A, *Technical Support and Professional Services*, for more information.
- Measurement Studio Web site, `ni.com/mstudio`—Contains Measurement Studio news, support, downloads, and evaluation software.
- NI Developer Zone, `ni.com/devzone`—Provides access to the latest online example programs, tutorials, and technical news, and a community of developers who are ready to share their techniques.
- *Measurement Studio .NET Class Hierarchy Chart* and *Measurement Studio Visual C++ Class Hierarchy Chart*—Provides an overview of class relationships within class libraries.

2

Measurement Studio .NET Class Libraries

This chapter provides overview information about the .NET class libraries included with Measurement Studio. Refer to the *Using the Measurement Studio .NET Class Libraries* section of the *NI Measurement Studio Help* for detailed information about these libraries.

Measurement Studio .NET Class Libraries Overview

Measurement Studio provides .NET class libraries that developers can use to develop complete measurement and automation applications in Visual Basic .NET and Visual C#.

Measurement Studio includes the following .NET class libraries:

- Analysis
- Common
- DataSocket
- NI-488.2
- NI-DAQmx
- NI-VISA
- User Interface

Refer to the following sections for information about each Measurement Studio .NET class library.

Analysis

The Measurement Studio Analysis .NET class library is in the `NationalInstruments.Analysis` namespace. The Analysis class library includes a set of classes that provides various digital signal processing, signal filtering, signal generation, peak detection, and other general mathematical functionality. Use this library to analyze data that you have acquired or to generate data.

The functionality included in the Analysis library varies based on the Measurement Studio package you purchased. Refer to the following sections for information about the Standard, Professional, and Enterprise Analysis class libraries.

Standard Analysis

The Standard Analysis class library, which ships with Measurement Studio Standard Edition, includes the following functions:

- Sawtooth wave generator
- Sine wave generator
- Square wave generator
- Triangle wave generator
- Basic function generator

Professional Analysis

Use the Professional Analysis class library, which ships with Measurement Studio Professional Edition, to perform the functions listed above, as well as the following operations:

- Create filters including Bessel, Chebyshev, Inverse Chebyshev, Windowed and Elliptic Low, High, Bandpass, and Bandstop.
- Perform signal processing using functions such as convolution, deconvolution, correlation, decimation, integration, and differentiation.
- Compute transforms such as FFT, Inverse FFT, Fast Hartley, Inverse Fast Hartley, Fast Hilbert, Inverse Fast Hilbert, Real FFT, and Laplace Real Transform.
- Use linear algebra functions such as check positive definiteness, determinant, calculate dot product, and various matrix methods.
- Use scaled and unscaled windowing classes.
- Use common statistical functions such as mean, median, mode, and variance.
- Use exponential, linear, and polynomial curve fitting.
- Perform signal generation.

Enterprise Analysis

Use the Enterprise Analysis class library, which ships with Measurement Studio Enterprise Edition, to perform the functions and operations listed above, as well as the following advanced operations:

- Create EquiRipple filters.
- Use linear algebra functions such as forward substitution, back substitution, and Cholesky factorization.
- Compute probability and analysis of variance.
- Generate patterns such as sinc, impulse, pulse, ramp, and chirp.
- Calculate general least square curve fit and use interpolation functions.

Common

The Measurement Studio Common .NET class library is in the `NationalInstruments` namespace. The Common class library provides a set of classes that facilitates the exchange of data between the acquisition, analysis, and user interface portions of your application. The Common class library includes the following features:

- A `ComplexDouble` data type. This data type represents a complex number of type Double that is composed of a real part and an imaginary part.
- A `DataConverter` class that converts data from one data type to another data type.

DataSocket

The Measurement Studio DataSocket .NET class library is in the `NationalInstruments.Net` namespace. Use the DataSocket class library to transfer live measurement data over the Internet or an intranet, between applications on the same computer, and to and from files. Use the classes in the DataSocket class library to perform the following operations:

- Read and write data between different data sources and targets.
- Use a single, simple API to communicate with several types of servers, including DataSocket Servers (`dstp:`), Web servers (`http:`), file transfer protocol servers (`ftp:`), file systems (`file:`), and OLE for Process Control (`opc:`) servers.
- Specify data sources and targets using a URL, the same way you access Web pages in a Web browser.

- Use DataSocket Transfer Protocol (DSTP) to exchange different types of data.
- Expose DataSocket data items as data sources that you can bind to properties of a Windows Forms control.
- Interactively browse to quickly locate and select data items on other computers and servers.

NI-488.2

The Measurement Studio NI-488.2 .NET class library is in the `NationalInstruments.NI4882` namespace. The NI-488.2 class library includes a set of classes that communicates with GPIB instruments, controls GPIB boards, and acquires GPIB status information. Use this library to design code that communicates with and controls instruments on a General Purpose Interface Bus (GPIB). Use the classes in the NI-488.2 class library to perform the following operations:

- Configure and communicate with GPIB instruments and devices.
- Perform I/O operations using the `Device` and `Board` classes.

You can use the NI-488.2 class library to create programs that interface with a device that is using GPIB and/or programs that interface with the GPIB directly.

Tip For information on easily creating a Measurement Studio 488.2 application using the NI Instrument I/O Assistant, refer to the *Creating a Measurement Studio 488.2 or VISA Application* section of Chapter 4, *Developing with Measurement Studio*.

NI-DAQmx

The Measurement Studio NI-DAQmx .NET class library is in the `NationalInstruments.DAQmx` namespace. Use the NI-DAQmx class library to communicate with and control an NI data acquisition (DAQ) device.

Note Some DAQ devices are not currently supported by the NI-DAQmx driver. Refer to the *NI-DAQ 7.0 Readme* for a complete listing of supported hardware.

Use the NI-DAQmx class library to perform the following types of tasks:

- Analog signal measurement
- Analog signal generation
- Digital I/O
- Counting and timing
- Pulse generation
- Signal switching

Tip For information on easily creating a DAQ application using the NI DAQ Assistant, refer to the *Creating a Measurement Studio DAQ Application* section of Chapter 4, *Developing with Measurement Studio*.

NI-VISA

The Measurement Studio NI-VISA .NET class library is in the `NationalInstruments.VisaNS` namespace. The NI-VISA class library includes a set of classes that provides a rich, object-oriented interface to the NI-VISA driver. Use this library to quickly create bus-independent and/or bus-specific instrument control applications.

The NI-VISA class library supports I/O operations, locking, event handling, and interface-specific extensions. With this class library you can access the functionality available in NI-VISA for communicating with message-based and register-based instruments using the following interfaces:

- GPIB
- PXI
- Serial (RS-232 and RS-485)
- TCP/IP
- VXI
- USB

Tip For information on easily creating a Measurement Studio VISA application using the NI Instrument I/O Assistant, refer to the *Creating a Measurement Studio 488.2 or VISA Application* section of Chapter 4, *Developing with Measurement Studio*.

User Interface

The Measurement Studio User Interface .NET class library is in the `NationalInstruments.UI.WindowsForms` namespace. The User Interface class library encapsulates the Measurement Studio user interface controls. Use this class library to add measurement-specific user interface controls to your application. Configure the controls programmatically or through the Properties window in the Windows Forms Designer. You also can configure the controls through various collection editors. The following sections describe each of the Measurement Studio user interface controls.

Scatter Graph and Waveform Graph Controls

Use the scatter graph and waveform graph controls, as shown in Figure 2-1, to display two-dimensional data on a Windows Forms user interface. The scatter graph control displays a graph of X and Y data pairs. The waveform graph control displays data that is uniformly spaced in one dimension.

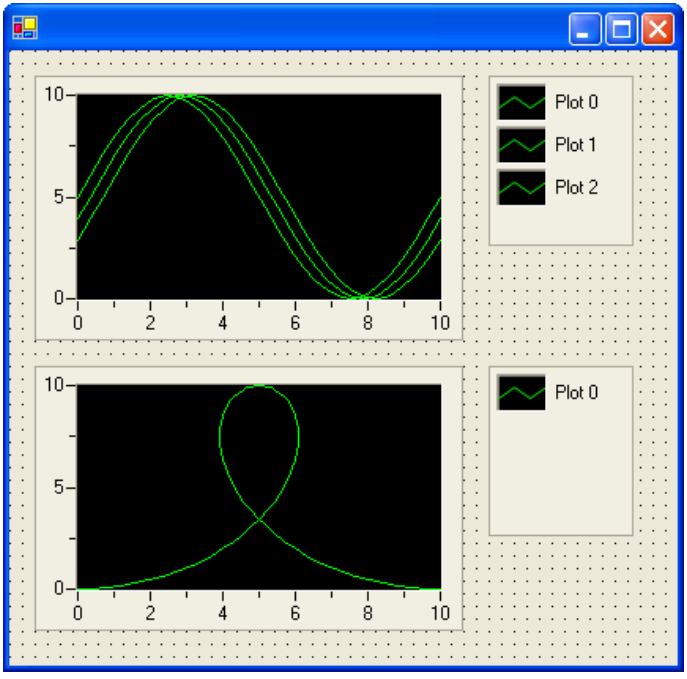

Figure 2-1. Waveform Graph and Scatter Graph with Corresponding Legends

The scatter graph and waveform graph controls and/or the classes that interface with the controls include the following major features:

- Plot and chart data.
- Configure a graph to contain multiple plots to show separate but related data on the same graph.
- Configure a graph to include multiple axes or independent ranges so that plot data fits the graph plot area.
- Specify plots in the scatter graph control as X and Y data. Specify plots in the waveform graph control as X or Y data and optionally with the date time and time span.
- Use cursors to identify key points in plots and the plot area.
- Configure cursor snap modes to be fixed, floating, nearest point, and to plot.
- Use the extensible plot and plot area drawing capabilities to customize the graph appearance.
- Use extensible point and line styles in plots and cursors.
- Pan and zoom interactively.
- Use NaN and positive/negative infinity support.
- Configure the axis modes to exact or loose autoscaling, fixed, strip chart, or scope chart.
- Configure major, minor, and custom divisions.
- Use logarithmic axes with a configurable base.

Switch and LED Controls

Use the switch and LED controls as boolean controls on a Windows Forms user interface. You typically use a switch control, as shown in Figure 2-2, to receive and control boolean input in an application user interface.

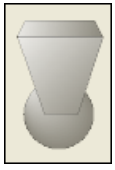

Figure 2-2. Switch Control

You typically use an LED control, as shown in Figure 2-3, to indicate a boolean value on an application user interface.

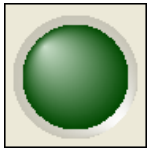

Figure 2-3. LED Control

The switch and LED controls and/or the classes that interface with the controls include the following major features:

- Receive notification when the current state of the control changes or before the change occurs.
- Configure how the control behaves when you click it with the mouse or press the spacebar when the control has focus.
- Configure the appearance of the control.
- Make the control background transparent.
- Configure the LED control to blink while it is on or off and configure the rate at which the LED control blinks.

Legend Control

Use the legend control, as shown in Figure 2-1, to display symbols and descriptions for a specific set of elements of another object, such as the plots or cursors of a graph. When you associate the legend control with another object, any changes you make to that object are automatically reflected in the legend. For example, if you associate the legend control with the plots of a graph, any changes you make in the plots collection editor are automatically reflected in the legend.

3

Measurement Studio Visual C++ Class Libraries

This chapter provides overview information about the Visual C++ class libraries that are available with Measurement Studio. Refer to the *Using the Measurement Studio Visual C++ Class Libraries* section of the *NI Measurement Studio Help* for detailed information about these libraries.

Measurement Studio Visual C++ Class Libraries Overview

Measurement Studio provides libraries of MFC-based classes that you can use to communicate with and control various National Instruments hardware products, component object model (COM) servers, and ActiveX controls. These Visual C++ class libraries offer a complete measurement solution that developers can use to develop robust measurement and automation applications in Visual C++ .NET.

Tip The DataSocket, NI-Reports, Microsoft Excel Interface, and Microsoft Word Interface class libraries communicate with COM servers. Refer to the *Microsoft Developer Network (MSDN) Library* for more information about COM servers.

Measurement Studio includes the following Visual C++ class libraries:

- 3D Graph
- Analysis
- Common
- DataSocket
- FieldPoint
- Instrument Drivers
- LabVIEW Real-Time Interface
- Microsoft Excel Interface
- Microsoft Word Interface

- NI-488.2
- NI-DAQ
- NI-Reports
- NI-VISA
- User Interface
- Utility

ActiveX Controls in Visual C++

ActiveX controls are specialized COM servers that implement a specific set of interfaces. The Measurement Studio Visual C++ User Interface and 3D Graph are ActiveX controls. The class libraries of these controls include classes that provide an interface to several ActiveX controls. For example, the `CNiGraph` class provides an interface to `CWUI.ocx` so that you can programmatically communicate with and control the ActiveX graph control.

The Measurement Studio classes that provide interfaces to the Measurement Studio ActiveX controls simplify using ActiveX controls in Visual C++ interfaces and programs. The features that simplify this process include overloaded functions, the ability to call the control from any thread, and automatic data type translations.

3D Graph

The Measurement Studio ActiveX 3D graph control and/or the classes that interface with the control include the following features:

- Plot three-dimensional data, including curves and surfaces. You can include three-dimensional axes and ticks on the plot.
- Configure the control to render directly to OpenGL-enabled hardware accelerator cards.
- Bind the control to a DataSocket Server to enable automatic read and write functionality.
- Use multiple plot styles—point-line, line-point, hidden-line, contour, surface, surface-line, surface-contour, and surface-normal.
- Create multiple plots with individual properties such as name, line and point style, width, and base value.
- Configure the axes using customizable ticks, labels, value pairs, and captions.

- Use legends and plane projections.
- Use cartesian, cylindrical, and spherical coordinate systems.
- Customize the control using color maps, transparency, and lighting.
- Display in orthographic and perspective views.
- Use built-in format styles for labels including scientific, symbolic engineering, scaling, time, and date.
- Use rotation, panning, and zooming at run time.

Analysis

The Measurement Studio Analysis Visual C++ class library includes a set of classes that provides various digital signal processing, signal filtering, signal generation, peak detection, and other general mathematical functionality. Use the Analysis Visual C++ class library to analyze data that you have acquired or to create data to generate.

The functionality included in the Analysis library varies based on the Measurement Studio package you purchased. Refer to following sections for information about the Standard, Professional, and Enterprise Analysis class libraries.

Standard Analysis

The Standard Analysis class library, which ships with Measurement Studio Standard Edition, includes the following signal generation functions:

- Sawtooth wave generator
- Sine wave generator
- Square wave generator
- Triangle wave generator

Professional Analysis

Use the Professional Analysis class library, which ships with Measurement Studio Professional Edition, to perform the following operations:

- Create filters such as Bessel, Chebyshev, Inverse Chebyshev, Windowed and Elliptic Low, High, Bandpass, and Bandstop.
- Perform signal processing using functions such as convolution, deconvolution, correlation, decimation, integration, and differentiation.

- Compute transforms such as FFT, Inverse FFT, Fast Hartley, Inverse Fast Hartley, Fast Hilbert, Inverse Fast Hilbert, Real FFT, and Laplace Real Transform.
- Use linear algebra functions such as check positive definiteness, determinant, calculate dot product, and various matrix methods.
- Use scaled and unscaled windowing classes.
- Use common statistical functions such as mean, median, mode, and variance.
- Use exponential, linear, and polynomial curve fitting.
- Perform signal generation.

Enterprise Analysis

Use the Enterprise Analysis class library, which ships with Measurement Studio Enterprise Edition, to perform the operations listed above, as well as the following advanced operations:

- Create EquiRipple filters.
- Use linear algebra functions such as forward substitution, back substitution, and Cholesky factorization.
- Compute probability and analysis of variance.
- Generate patterns such as sinc, impulse, pulse, ramp, and chirp.
- Calculate general least square curve fit and use interpolation functions.

Common

The Measurement Studio Common Visual C++ class library provides data types and classes that other Measurement Studio Visual C++ class libraries use. The classes that are implemented natively in Visual C++ include the `CNiVector` and `CNiMatrix` classes.

The Common class library includes the following data types:

- `CNiScalarVector`—Implements a vector object that contains scalar numbers.
- `CNiScalarMatrix`—Implements a matrix object that contains scalar numbers.
- `CNiString`—Extends the MFC `CString` class with streaming operators for a variety of data types and with various other string manipulation functions.

- `CNiVariant`—Extends the MFC `COleVariant` class with additional constructors and assignment operators for `CNiComplex`, `CNiVector`, and `CNiMatrix` derived objects and with cast operators to convert `CNiVariant` objects to a variety of other object types.
- `CNiException`—Extends the MFC `CException` class and serves as the base class for many Measurement Studio exceptions.
- `CNiRegKey`—Encapsulates the interface to the Windows registry. Use this class and related classes to open and create keys, get keys, and get values associated with those keys.

DataSocket

Use the Measurement Studio DataSocket Visual C++ class library to transfer live measurement data over the Internet or an intranet, between applications on the same computer, and to and from files. Use the classes in the DataSocket Visual C++ class library to perform the following operations:

- Read and write data between different data sources and targets.
- Use a single, simple API to communicate with several types of servers, including DataSocket Servers (`dstp:`), Web servers (`http:`), file transfer protocol servers (`ftp:`), file systems (`file:`), and OLE for Process Control (`opc:`) servers.
- Specify data sources and targets using a URL, the same way you access Web pages in a Web browser.
- Use DataSocket Transfer Protocol (DSTP) to exchange different types of data.
- Interactively browse to quickly locate and select data items on other computers and servers.

FieldPoint

Use the Measurement Studio FieldPoint Visual C++ class library to design Visual C++ applications that communicate with National Instruments FieldPoint modules. The FieldPoint Visual C++ class library provides programmatic access to all FieldPoint configuration options over Ethernet.

Use the classes in the FieldPoint Visual C++ class library to perform the following operations:

- Automatically detect which modules and channels are available.
- Install event handlers for asynchronous notifications of channel data changes.
- Integrate Measurement Studio wizards, classes, and help documentation.

National Instruments FieldPoint is a modular distributed I/O system for industrial applications. For more information, visit `ni.com/fieldpoint`.

Instrument Drivers

Measurement Studio includes a set of class libraries that provides Visual C++ classes that you can use to program instruments, such as DMMs and oscilloscopes. These Visual C++ classes provide native C++ interfaces to C DLL-based IVI instrument drivers. Each Measurement Studio instrument driver class library includes a set of C++ classes that interfaces to a single IVI driver. The Measurement Studio C++ classes provide enhancements to the C interface such as automatic data type translation and organization of properties and methods into a logical hierarchy.

IVI instrument drivers are divided into two main categories—class drivers and specific drivers. Each class driver provides a consistent interface to particular types of devices. Measurement Studio includes the following class driver class libraries:

- IviDcPwr—Provides an interface to the IVI DC power supply class driver. `CNiIviDcPwr` is the top-level class of this class library.
- IviDmm—Provides an interface to the IVI DMM class driver. `CNiIviDmm` is the top-level class of this class library.
- IviFgen—Provides an interface to the IVI arbitrary waveform generator class driver. `CNiIviFgen` is the top-level class of this class library.
- IviScope—Provides an interface to the IVI oscilloscope class driver. `CNiIviScope` is the top-level class of this class library.
- IviSwitch—Provides an interface to the IVI switch class driver. `CNiIviSwtch` is the top-level class of this class library.

Each specific driver provides an interface to a particular device, such as the NI-DMM modular instrument. National Instruments provides Measurement Studio Visual C++ instrument driver class libraries for

the National Instruments modular instruments and for many third-party vendor instruments. You can download the Measurement Studio Visual C++ instrument driver class libraries from `ni.com/idnet`.

LabVIEW Real-Time Interface

Use the Measurement Studio LabVIEW Real-Time Interface Visual C++ class library to read from and write to shared memory on a LabVIEW RT Series processor board. Use shared memory to pass data between LabVIEW RT VIs and your application. Use this class library both from an application that runs on the host machine and from a DLL that you download to the board.

For more information about LabVIEW RT, visit `ni.com/labviewrt`.

Note The LabVIEW RT DLLs you create with the Measurement Studio LabVIEW RT Interface Visual C++ class library work only with the LabVIEW RT software version 6.0 or later.

Microsoft Excel Interface

Use the Measurement Studio Excel Visual C++ class library to automatically create Excel spreadsheets and charts from within measurement and automation applications. Use the Microsoft Excel Interface class library to perform offline processing of the measurement and automation data you acquire and analyze using other Measurement Studio Visual C++ classes.

Microsoft Word Interface

Use the Measurement Studio Microsoft Word Interface Visual C++ class library to automatically create Word documents from within measurement and automation applications. Use the Microsoft Word Interface class library to perform offline processing of the measurement and automation data you acquire and analyze using other Measurement Studio Visual C++ classes.

NI-488.2

Use the Measurement Studio NI-488.2 Visual C++ class library to communicate with and control instruments on a General Purpose Interface Bus (GPIB). Use the classes in this library to perform the following operations:

- Perform I/O with GPIB instruments and devices.
- Configure GPIB instruments and devices.
- Perform event handling for GPIB instruments and devices.

You can use the NI-488.2 class library to create programs that interface with a device that is using GPIB and/or programs that interface with the GPIB directly.

Tip For information on easily creating a Measurement Studio 488.2 application using the NI Instrument I/O Assistant, refer to the *Creating a Measurement Studio 488.2 or VISA Application* section of Chapter 4, *Developing with Measurement Studio*.

NI-DAQmx

Use the Measurement Studio NI-DAQmx Visual C++ class library to communicate with and control an NI data acquisition (DAQ) device.

Note Some DAQ devices are not currently supported by the NI-DAQmx driver. Refer to the *NI-DAQ 7.0 Readme* for a complete listing of supported hardware.

Use the NI-DAQmx class library to perform the following types of tasks:

- Analog signal measurement
- Analog signal generation
- Digital I/O
- Counting and timing
- Pulse generation
- Signal switching

Tip For information on easily creating a DAQ application using the NI DAQ Assistant, refer to the *Creating a Measurement Studio DAQ Application* section of Chapter 4, *Developing with Measurement Studio*.

NI-Reports

Use the Measurement Studio NI-Reports Visual C++ class library to generate printed reports from Measurement Studio Visual C++ applications.

NI-VISA

The Measurement Studio NI-VISA Visual C++ class library includes Visual C++ classes that provide an object-oriented interface to the NI-VISA driver. Use the NI-VISA class library to quickly create bus-independent and/or bus-specific instrument control applications.

The NI-VISA class library supports I/O operations, locking, event handling, and interface-specific extensions. With this class library you can access the functionality available in NI-VISA for communicating with message-based and register-based instruments using the following interfaces:

- GPIB
- PXI
- Serial (RS-232 and RS-485)
- TCP/IP
- USB
- VXI

Tip For information on easily creating a Measurement Studio VISA application using the NI Instrument I/O Assistant, refer to the *Creating a Measurement Studio 488.2 or VISA Application* section of Chapter 4, *Developing with Measurement Studio*.

User Interface

Use the Measurement Studio User Interface Visual C++ class library to add user interface controls to your application. You can configure the user interface controls programmatically or through the property pages in the Visual C++ resource editor. The following sections describe each of the Measurement Studio Visual C++ user interface controls.

Button

Use the Measurement Studio ActiveX button control for different boolean displays, such as on/off or true/false. Typically, you use buttons to input or display boolean information or initiate an action in a program. The `CNiButton` class provides the Visual C++ interface to the ActiveX button control.

The button control and/or the classes that interface with the control include the following major features:

- Configure how the control behaves when you click it with the mouse or press the spacebar when the control has focus.
- Determine how the button control appears using button styles. You can configure the button control to appear as a push button, LED, or switch.
- Bind properties to a DataSocket source or target. You use binding to read property values from a source and write property values to a target.

Graph

Use the Measurement Studio ActiveX graph control to plot and chart two-dimensional data. The `CNiGraph` class provides the Visual C++ interface to the ActiveX graph control.

The graph control and/or the classes that interface with the control include the following major features:

- Plot and chart data.
- Configure a graph to contain multiple plots to show separate but related data on the same graph.
- Configure a graph to include multiple axes so that plot data fits the graph plot area.
- Use cursors and annotations to identify key points in plots and the plot area.
- Configure cursor snap modes to be fixed, floating, nearest point, and to plot.
- Pan and zoom interactively.
- Use the `CNiAxis` class to interface to a single axis of a graph control. This feature allows you to modify the appearance and behavior of the axis.

- Automatically label axes with log or inverted numeric scales and continuous or discrete values.
- Configure the axis modes to exact or loose autoscaling, fixed, strip chart, or scope chart.
- Customize the graph by using ticks, labels, and value pairs.
- Bind properties to a DataSocket source or target. You use binding to read property values from a source and write property values to a target.

Knob

Use the Measurement Studio ActiveX knob control for different types of circular displays of numerical information. The `CNiKnob` class provides the Visual C++ interface to the ActiveX knob control.

The knob control and/or the classes that interface with the control include the following major features:

- Use different display styles—dials, gauges, and meters.
- Use multiple control pointers, each representing one scalar value. A control pointer indicates the current value of the knob.
- Use the `CNiAxis` class to interface to a single axis of a knob control. This feature allows you to modify the appearance and behavior of the axis.
- Automatically label axes with log or inverted numeric scales and continuous or discrete values.
- Customize the knob by using ticks, labels, and value pairs.
- Bind properties to a DataSocket source or target. You use binding to read property values from a source and write property values to a target.

Numeric Edit

Use the Measurement Studio ActiveX numeric edit control to display numeric values and provide a method by which end users can edit numeric values. Typically, you use a numeric edit control to input or display numerical data instead of using a text box. The `CNiNumEdit` class provides the Visual C++ interface to the ActiveX numeric edit control.

The numeric edit control and/or the classes that interface with the control include the following major features:

- Perform range checking.
- Use built-in numeric format styles, including scientific, symbolic engineering, scaling, time, and date.
- Bind properties to a DataSocket source or target. You use binding to read property values from a source and write property values to a target.

Slide

Use the Measurement Studio ActiveX slide control for different types of linear displays of numerical data. `CNiSlide` is the class that provides the Visual C++ interface to the ActiveX slide control.

The slide control and/or the classes that interface with the control include the following major features:

- Use different display styles including vertical, horizontal, and thermometer.
- Use the `CNiAxis` class to interface to a single axis of a slide control. This ability allows you to modify the appearance and behavior of the axis.
- Use multiple control pointers, each one representing one scalar value. A control pointer indicates the current value of the slide.
- Automatically label axes with log or inverted numeric scales and continuous or discrete values.
- Customize the slide by using ticks, labels, and value pairs.
- Animate different parts of the control.
- Bind properties to a DataSocket source or target. You use binding to read property values from a source and write property values to a target.

Utility

Use the Measurement Studio Utility Visual C++ class library to easily access Windows operating system (OS) functionality. The following sections describe the classes that make up the Utility class library and a brief description of the class functionality.

CNiFile

`CNiFile` extends the MFC `CStdioFile` class by adding streaming operators for standard C++ data types. In addition, a variety of class static functions add the ability to manipulate file, path, directory, and drive attributes.

CNiSound

`CNiSound` encapsulates an interface for generating synchronous and asynchronous tones at specific frequencies.

CNiSystem

`CNiSystem` provides the following functionalities:

- Getting and setting system preferences
- Displaying help files
- Getting input for the keyboard

CNiSystemTrayIcon

`CNiSystemTrayIcon` encapsulates the interface to the system tray area that displays changes in the status of an application. The `CNiSystemTrayIcon` class includes the following features:

- Icons—You can place an icon in the system tray to notify the user of changes in an application status.
- String tool tips—You can associate a string tool tip with an icon and display the tool tip when the user hovers over the icon.
- Shortcut menus—You can associate a shortcut menu with an icon and display the shortcut menu when the user right-clicks the icon.
- Overridable event handling.

CNiTempFile

`CNiTempFile` extends the functionary of `CNiFile` to add temporary file capability.

CNiTimer

`CNiTimer` objects use the Windows multimedia timer to generate high-resolution, asynchronous tick events. Respond to tick events when you want to perform an action at a discrete interval. Additionally, you can count the tick events to calculate elapsed time. The `CNiTimer` class also contains static functions you can use to delay for a period of time or to determine elapsed time between two points in your program.

4

Developing with Measurement Studio

When you use Measurement Studio in the Visual Studio .NET environment, you have access to measurement and automation tools and features created with Visual Basic .NET, Visual C#, and Visual C++ .NET. These integrated tools and features are designed to help you quickly and easily build measurement and automation applications.

This chapter includes the following sections to help you develop applications with Measurement Studio:

- *Measurement Studio Menu*
- *Creating a New Measurement Studio Project*
- *Creating a Measurement Studio DAQ Application*
- *Creating a Measurement Studio 488.2 or VISA Application*
- *Adding or Removing Measurement Studio Class Libraries*
- *Selecting a Measurement Studio Parameter Value*

These sections include overview information on developing applications with Measurement Studio tools and features. Refer to the *Developing with Measurement Studio* section of the *NI Measurement Studio Help* for more information about the functionality of these tools and features.

Measurement Studio Menu

The Measurement Studio Menu provides an easy way to access important National Instruments resources and tools. The Measurement Studio Menu contains links to the following resources:

- **Add/Remove Class Libraries Wizard**—Use the Measurement Studio Add/Remove Class Libraries wizard to add or remove Measurement Studio class libraries or assemblies from an existing Visual Basic .NET, Visual C#, or Visual C++ .NET project.

- **Measurement Studio Preferences**—Use the Measurement Studio Preferences dialog box to configure settings in Measurement Studio, such as the conversion options, add-in preferences, and reset toolbox.
- **NI Measurement Studio Help**—Use the *NI Measurement Studio Help* to access detailed Measurement Studio help, including function reference and in-depth documentation on developing with Measurement Studio.
- **Parameter Assistant**—Use the Measurement Studio Parameter Assistant to discover and insert valid parameter values for DataSocket, DAQ, GPIB, and VISA methods.
- **Measurement & Automation Explorer (MAX)**—Use MAX to configure NI hardware; add new channels, interfaces, and virtual channels; execute system diagnostics; and view devices and instruments connected to the system.
- **NI Spy**—Use NI Spy to monitor, record, and display National Instruments API calls made by instrument connectivity applications. Use NI Spy to quickly locate and analyze any erroneous National Instruments API calls that an application makes and verify that the communication with an instrument is correct.
- **Developer Exchange**—Use NI Developer Exchange at `ni.com/devzone` to participate in discussion forums and exchange code with measurement and automation developers around the world.
- **Search Technical Support**—Use NI Technical Support at `ni.com/support` to find support resources available for most products at no cost to registered users, including software drivers and updates, a KnowledgeBase, product manuals, step-by-step troubleshooting wizards, conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, and a measurement glossary.
- **Measurement Studio Home**—Use the Measurement Studio Web site at `ni.com/mstudio` to find Measurement Studio news, support, downloads, and evaluation software.
- **Instrument Driver Network**—Use the NI Instrument Driver Network at `ni.com/idnet` as a central resource for downloading, developing, and submitting instrument drivers.
- **Measurement Encyclopedia**—Use the online NI Measurement Encyclopedia to find information on measurement principles, standards organizations, and a wide range of technology and measurement terms.

- **About NI Measurement Studio**—Use the NI Measurement Studio About box for version information.
- **Patents**—Use the Patents dialog box for information about NI patents.

For more information about the resources included in the Measurement Studio Menu, refer to the *Measurement Studio Menu* section of the *NI Measurement Studio Help*.

Creating a New Measurement Studio Project

Measurement Studio includes class library and application templates that you can use to quickly create measurement applications with Visual Basic .NET, Visual C#, and Visual C++ .NET. Use the Visual Studio .NET New Project dialog box, as shown in Figure 4-1, to access these templates and create the following types of projects:

- Measurement Studio Visual Basic .NET project
- Measurement Studio Visual C# project
- Measurement Studio Visual C++ .NET project
- Measurement Studio Visual C++ project with LabWindows™/CVI™ libraries

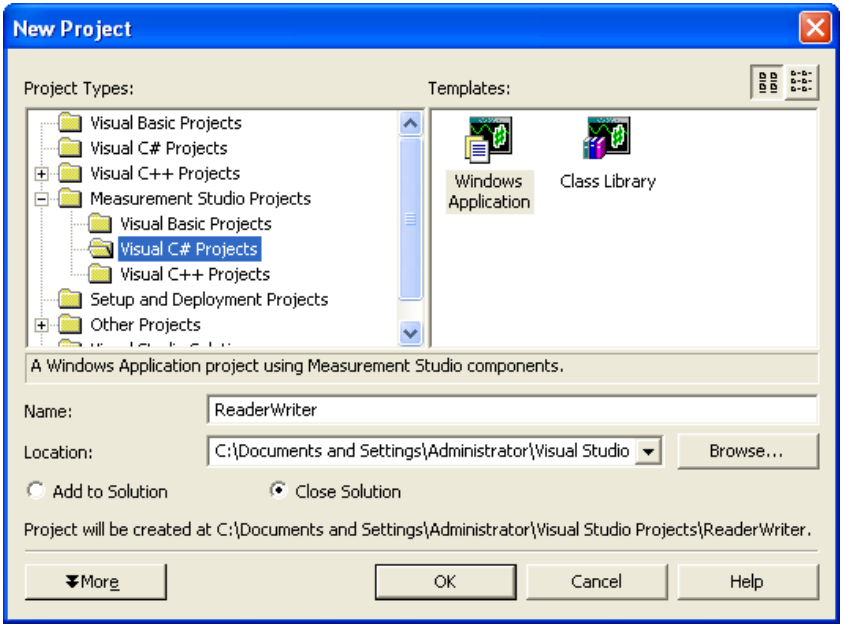

Figure 4-1. New Project Dialog Box

For more information about using project templates to create a new Measurement Studio project, refer to the *Creating a New Measurement Studio Project* section of the *NI Measurement Studio Help*.

For information about converting Measurement Studio projects, refer to the *Converting Measurement Studio Projects* section of the *NI Measurement Studio Help*.

Creating a Measurement Studio DAQ Application

To create a Measurement Studio DAQ application, use the NI DAQ Assistant. The DAQ Assistant integrates into Visual Studio .NET as a code designer. Use the DAQ Assistant user interface, as shown in Figure 4-2, to create and configure DAQ tasks. The DAQ Assistant automatically generates a Visual Basic .NET, Visual C#, or Visual C++ .NET class that includes the functionality you configure in the user interface.

The DAQ Assistant interactively assists you in performing the following operations:

- Creating a DAQmx task class
- Configuring a DAQmx task class
- Generating a Visual Basic .NET, Visual C#, or Visual C++ .NET class that includes the functionality you configure in the user interface
- Generating code that uses a DAQmx task class
- Using a DAQmx task class in a project

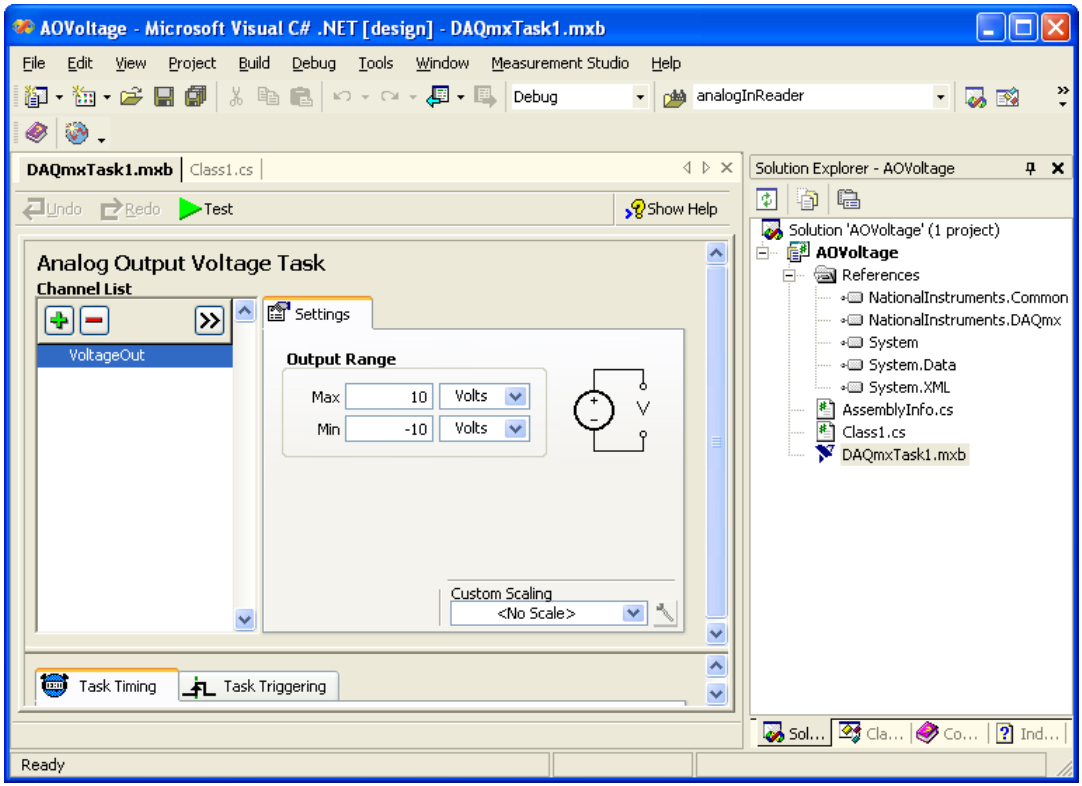

Figure 4-2. The DAQ Assistant

For more information about using the DAQ Assistant to create a Measurement Studio DAQ application, refer to the *Creating a Measurement Studio DAQ Application* section of the *NI Measurement Studio Help*.

Creating a Measurement Studio 488.2 or VISA Application

To create a Measurement Studio 488.2 or VISA application, use the NI Instrument I/O Assistant. The Instrument I/O Assistant integrates into Visual Studio .NET as a code designer. Use the Instrument I/O Assistant user interface to create and configure instrumentation tasks. The Instrument I/O Assistant generates a Visual Basic .NET, Visual C#, or Visual C++ class that includes the functionality you configure in the user interface. Use this assistant to help you write code that communicates with devices such as serial, Ethernet, or GPIB instruments.

The Instrument I/O Assistant assists you in performing the following operations:

- Creating an instrumentation task class
- Configuring an instrumentation task class
- Generating a Visual Basic .NET, Visual C#, or Visual C++ class that includes the functionality you configure in the user interface

For more information about using the NI Instrument I/O Assistant to create a Measurement Studio 488.2 or VISA application, refer to the *Creating a Measurement Studio 488.2 or VISA Application* section of the *NI Measurement Studio Help*.

Adding or Removing Measurement Studio Class Libraries

To add or remove Measurement Studio class libraries, use the Measurement Studio Add/Remove Class Libraries wizard. This wizard provides an interface, as shown in Figure 4-3, that you can use to select the Measurement Studio class libraries or assemblies you want to add or remove from a project.

When you exit the wizard, the wizard adds or removes the appropriate references to or from the project, thus adding or removing the functionality associated with the class library.

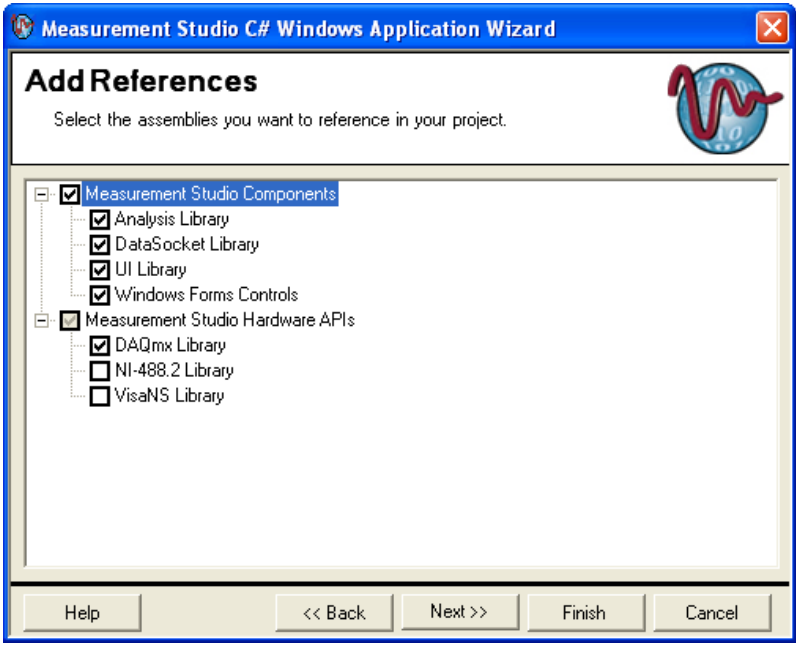

Figure 4-3. The Measurement Studio Add/Remove Class Libraries Wizard

For more information about using the Add/Remove Class Libraries wizard to add or remove Measurement Studio class libraries, refer to the *Adding or Removing Measurement Studio Class Libraries* section of the *NI Measurement Studio Help*.

Selecting a Measurement Studio Parameter Value

To access devices or resources, you must specify string constants or scalar values for many method parameters. Use the Measurement Studio Parameter Assistant to discover and insert into your code valid parameter values for DataSocket, DAQ, GPIB, and VISA methods.

With the Parameter Assistant, you can select the correct parameter value for a device or resource, as shown in Figure 4-4, based on your current system configuration. Click a button on the Parameter Assistant to insert the value into the current location in the active source file.

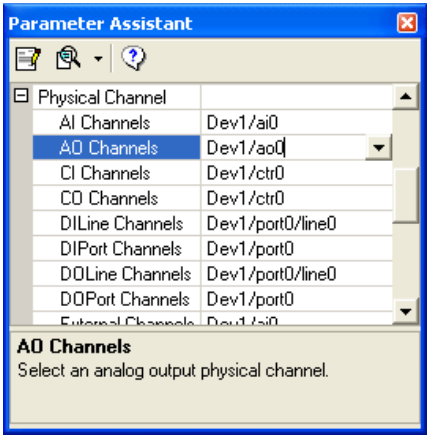

Figure 4-4. The Measurement Studio Parameter Assistant

For more information about using the Measurement Studio Parameter Assistant to select a parameter value, refer to the *Selecting a Measurement Studio Parameter Value* section of the *NI Measurement Studio Help*.

A

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- **Support**—Online technical support resources include the following:
  - **Self-Help Resources**—For immediate answers and solutions, visit our extensive library of technical support resources available in English, Japanese, and Spanish at ni.com/support. These resources are available for most products at no cost to registered users and include software drivers and updates, a KnowledgeBase, product manuals, step-by-step troubleshooting wizards, conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, a measurement glossary, and so on.
  - **Assisted Support Options**—Contact NI engineers and other measurement and automation professionals by visiting ni.com/support. Our online system helps you define your question and connects you to the experts by phone, discussion forum, or email.
- **Training**—Visit ni.com/training for self-paced tutorials, videos, and interactive CDs. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, NI Alliance Program members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Glossary

A

ActiveX	Set of Microsoft technologies for reusable software components. Formerly called OLE.
ActiveX control	Reusable software component that adds functionality to any ActiveX control container through exposed properties, methods, and events. The Measurement Studio data acquisition, user interface, and analysis 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.
analog I/O	Reading or writing data in continuously variable physical quantities, such as voltage or current.
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.
asynchronous	Function that begins an operation and returns control to the program prior to the completion or termination of the operation.

C

channel	1. Physical—a terminal or pin at which you can measure or generate an analog or digital signal. A single physical channel can include more than one terminal, as in the case of a differential analog input channel or a digital port of eight lines. The name used for a counter physical channel is an exception because that physical channel name is not the name of the terminal where the counter measures or generates the digital signal.

	2. Virtual—a collection of property settings that can include a name, a physical channel, input terminal connections, the type of measurement or generation, and scaling information. You can define NI-DAQmx virtual channels outside a task (global) or inside a task (local). Configuring virtual channels is optional in Traditional NI-DAQ and earlier versions, but is integral to every measurement you take in NI-DAQmx. In Traditional NI-DAQ, you configure virtual channels in MAX. In NI-DAQmx, you can configure virtual channels in either MAX or in a program, and you can configure channels as part of a task or separately. 3. Switch—a switch channel represents any connection point on a switch. It may be made up of one or more signal wires (commonly one, two, or four), depending on the switch topology. A virtual channel cannot be created with a switch channel. Switch channels may be used only in the NI-DAQmx Switch functions and VIs.
chart	To append new data points to the end of an existing plot over time.
CodeBuilder	LabWindows/CVI feature that creates code based on a `.uir` 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 specification for architecting and developing reusable software components.
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 **Help** button, or by clicking the link that appears in the Dynamic Help window in Visual Studio .NET.
control	1. ActiveX control. *See* ActiveX control. 2. Object for entering or manipulating data on a user interface. Compare with indicator.
counter/timer I/O	Reading or writing data based on high-precision timing through a counter or timer. By combining a counter with a highly accurate clock, you can create a wide variety of timing and counting applications, such as monitoring and analyzing digital waveforms and generating complex square waves.
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.

D

DAQ	Data acquisition. Process of acquiring data, typically from A/D or digital input plug-in boards.
DAQ Assistant	A graphical interface for configuring measurement tasks, channels, and scales.
DAQ device	A device that acquires or generates data and can contain multiple channels and conversion devices. DAQ devices include plug-in devices, PCMCIA cards, and DAQPad devices, which connect to a computer USB or 1394 (FireWire) port. SCXI modules are considered DAQ devices.
DataSocket	Technology that simplifies live data exchange between applications and HTTP, FTP, OPC, logos (Lookout objects), and file servers over the Internet. It provides one common API to a number of different communication protocols.
device	An instrument or controller you can access as a single entity that controls or monitors real-world I/O points. A device is often connected to a host computer through some type of communication network. *See also* DAQ device and measurement device.
digital I/O	Reading or writing digital representations of data in discrete units (the binary digits 1 and 0). Digital information is either on or off.
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 run time 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	Software that controls a specific hardware device, such as a data acquisition board or GPIB interface board. *See also* instrument driver.
DSTP	DataSocket Transfer Protocol. Protocol based on TCP/IP to exchange data directly between two applications using DataSocket clients. Data is passed through a DataSocket Server between the applications.

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 that processes the event.
executable	Program file with a `.exe` extension that you can run independently of the development environment in which it was created.

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.
FTP	File Transfer Protocol. Protocol based on TCP/IP to exchange files between computers.

G

GPIB	General Purpose Interface Bus. The standard bus used for controlling electronic instruments with a computer. Also called IEEE 488 bus because it is defined by ANSI/IEEE Standards 488-1978, 488.1-1987, and 488.2-1987.

H

HTTP	HyperText Transfer Protocol. Protocol based on TCP/IP, which is used to download Web pages from an HTTP server to a Web browser.

I

IEEE 488	Shortened notation for ANSI/IEEE Standards 488-1978, 488.1-1987, and 488.2-1987. *See also* GPIB.
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. Compare with control.
installer	Software program that copies program, system, and other necessary files to computers.
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.
Instrument I/O Assistant	Assists in writing code to communicate with devices such as serial, Ethernet, or GPIB instruments. The Instrument I/O Assistant provides a user interface within the Visual Studio .NET environment. You use the Instrument I/O Assistant user interface to interactively write commands to a device, read data that the device returns, and specify how to parse the response.
interface	Connection between one or more of the following: hardware, software, and the user. For example, hardware interfaces connect two other pieces of hardware.
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 `www.ivifoundation.org`.

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	NI 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.

M

MB	Megabytes of memory.
Measurement & Automation Explorer (MAX)	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 device	DAQ devices such as the E Series multifunction I/O (MIO) devices, SCXI signal conditioning modules, and switch modules.
Measurement Studio	National Instruments software that includes tools to build measurement applications in Visual Basic .NET, Visual C#, and Visual C++ .NET.
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.

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 DAQ hardware. NI-DAQ is an extensive library of VIs and functions you can call from an application development environment (ADE) to program all the features of an NI measurement device, such as configuring, acquiring and generating data from, and sending data to the device.
NI-DAQmx	The latest NI-DAQ driver with new VIs, functions, and development tools for controlling measurement devices. The advantages of NI-DAQmx over earlier versions of NI-DAQ include the DAQ Assistant for configuring channels and measurement tasks for your device for use in LabVIEW, LabWindows/CVI, and Measurement Studio; increased performance such as faster single-point analog I/O; and a simpler API for creating DAQ applications using fewer functions and VIs than earlier versions of NI-DAQ.
NI-IMAQ	Driver-level software to control and communicate with National Instruments image acquisition hardware.

O

OCX	OLE Control eXtension. Another name for ActiveX controls, reflected by the `.ocx` file extension of ActiveX control files.
OLE	Object Linking and Embedding. *See* ActiveX.
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`.
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.
plot	1. Trace (data line) on a graph representing the data in one row or column of an array. 2. To display a new set of data while deleting any previous data on the graph.
property	Attribute that defines the appearance or state of an object. The property can be a specific value or another object with its own properties and methods. For example, a value property is the color (property) of a plot (object), while an object property is a specific Y axis (property) on a graph (object). The Y axis itself is another object with properties, such as minimum and maximum values.
property pages	Window or dialog box that displays current configuration information and allows users to modify the configuration.
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.
synchronous	Property or operation that begins and returns control to the program only when the operation is complete.

T

task	NI-DAQmx—a collection of one or more channels, timing, and triggering and other properties that apply to the task itself. Conceptually, a task represents a measurement or generation you want to perform.
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.

V

virtual instrument (VI)	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 VXI*plug&play* Systems Alliance.
VXI	VME eXtension for Instrumentation. Instrumentation architecture and bus based on the VME standard. Used in high-end test applications.

Index

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W