

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: 512 683 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 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™, CVI™, DAQPad™, DataSocket™, FieldPoint™, IMAQ™, IVI™, LabVIEW™, Lookout™, Measurement Studio™, National Instruments™, NI™, NI Developer Suite™, NI Developer Zone™, ni.com™, NI-488.2™, NI-DAQ™, NI-DMM™, NI-IMAQ™, NI-VISA™, SCXI™, and TestStand™ are trademarks of National Instruments Corporation.

FireWire® is the trademark of Apple Computer, Inc., registered in the U.S. and other countries. Other product and company names mentioned herein are trademarks or trade names of their respective companies.

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.

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 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.

Contents

Chapter 1

Virtual Instrumentation

NI Platform	1-1
Virtual Instrumentation.....	1-1
Acquire, Analyze, and Present.....	1-2
Acquire	1-2
Analyze.....	1-2
Present	1-3
Measurement Studio	1-3
.NET Class Libraries	1-3
Visual C++ Class Libraries	1-3
Measurement Studio Tools.....	1-4
Related Software Packages	1-4
NI Developer Suite	1-4
TestStand	1-4
Vision and Image Processing Software.....	1-5
Motion Control Module.....	1-5
Automation Symbols Toolset.....	1-5
PID Control Toolset	1-5

Chapter 2

Getting Started with Measurement Studio

About This Evaluation Package.....	2-1
System Requirements	2-1
Installation Instructions.....	2-2
Using Measurement Studio Help.....	2-3
Context-Sensitive Help.....	2-4

Chapter 3

What is Measurement Studio?

Why Should I Use Measurement Studio?.....	3-1
Intuitive Measurement Hardware APIs and Assistants	3-1
Extensible Measurement Classes	3-2
Native .NET User Interface Controls	3-2
Advanced Analysis Libraries	3-2
Industry-Leading Software Expertise.....	3-2
Microsoft Visual Studio Integration	3-3

Chapter 4

Developing Your Application with Measurement Studio

Getting Started.....	4-1
Creating a New Measurement Studio Project.....	4-2
Adding or Removing Measurement Studio Class Libraries	4-4
Selecting a Measurement Studio Parameter Value	4-5
Measurement Studio .NET Class Libraries	4-6
Acquire.....	4-7
Analyze	4-8
Present.....	4-8
Networking.....	4-9
Measurement Studio Visual C++ Class Libraries	4-10
Acquire.....	4-10
Analyze	4-13
Present.....	4-13
Networking.....	4-14
Distributing a Measurement Studio Application	4-14

Chapter 5

Where To Go From Here

Related Documentation	5-1
Measurement Studio Examples	5-2

Appendix A

Technical Support and Professional Services

Glossary

Virtual Instrumentation

NI Platform

National Instruments is committed to providing software and hardware for engineers and scientists who develop 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.

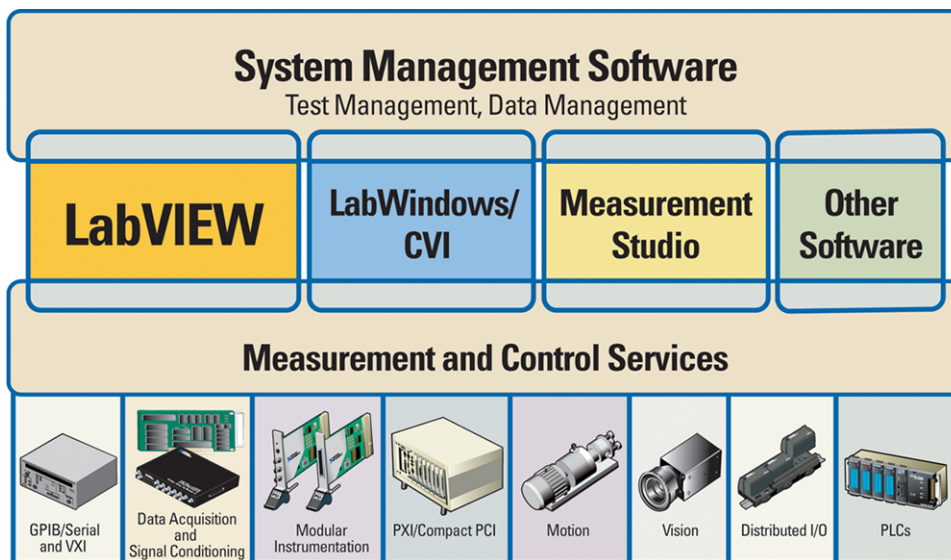


Figure 1-1. NI Platform

Virtual Instrumentation

Virtual instruments represent a fundamental shift from traditional hardware-centered instrumentation systems to software-centered systems that exploit the computing power, productivity, display, and connectivity capabilities of popular desktop computers and workstations. Although the PC and integrated circuit technology have experienced significant

advances in the last two decades, it is software that truly provides the leverage to build on this powerful hardware foundation to create virtual instruments, providing better ways to innovate and significantly reduce cost. With virtual instruments, engineers and scientists build user-defined measurement and automation systems that suit their needs exactly, instead of being limited by traditional vendor-defined instruments.

Software is the cornerstone of a virtual instrumentation system. Its flexibility, combined with powerful modular hardware solutions, creates the ultimate in user-defined, scalable instrumentation systems. Measurement Studio is the software tool for creating virtual instruments with Microsoft Visual Studio .NET. Measurement Studio adds powerful, intuitive measurement and control-specific development tools to standard Microsoft development tools to increase your efficiency in creating true virtual instruments.

Acquire, Analyze, and Present

Measurement Studio is a productive suite of measurement and automation tools and class libraries for Microsoft Visual C#, Visual Basic .NET, and Visual C++. These development tools help engineers and scientists rapidly implement the fundamentals of any measurement application—acquisition, analysis, and presentation.

Acquire

Measurement Studio delivers seamless connectivity with measurement hardware so you can quickly configure and use virtually any measurement device, including everything from stand-alone instruments to plug-in data acquisition devices, motion controllers, image acquisition systems, and programmable logic controllers (PLCs). Plus, Measurement Studio works with more than 1,000 instrument libraries from hundreds of vendors.

Analyze

Measurement Studio provides powerful algorithms and functions designed specifically for measurement analysis and signal processing. You can easily integrate these functions into Visual Studio .NET to create robust virtual instruments by transforming raw data into critical information. The comprehensive .NET and Visual C++ libraries for measurement analysis, signal processing, and mathematical analysis extract information from acquired data and unique measurements, generate, modify, process, and analyze signals, and add intelligence and decision-making capabilities to applications.

Present

Extensible .NET user interface (UI) controls deliver unparalleled flexibility through customizable features. This flexibility, combined with measurement-specific UI controls, such as 2D graphs, switches, and LED controls, empowers the creation of intuitive front panels for virtual instrument applications.

Measurement Studio also offers ActiveX UI controls—real-time 2D and 3D graphs, knobs, meters, gauges, tanks, and switches—for Visual C++. Measurement Studio presents its virtual instrumentation controls seamlessly in Visual C++, as multithread-safe MFC controls.

In addition to professional presentation capabilities, you can use National Instruments DataSocket to easily bind any user interface control, including NI controls and native Visual Studio controls, to live measurement data without writing code.

Measurement Studio

Measurement Studio offers development tools and rich, object-oriented classes for Visual C#, Visual Basic .NET, and Visual C++ programmers.

.NET Class Libraries

Measurement Studio provides a suite of native class libraries designed for engineers and scientists building virtual instrumentation systems with Visual Studio .NET. Measurement Studio includes simplified data networking and native .NET user interface controls. Advanced analysis libraries and rich object-oriented hardware APIs, such as data acquisition and instrument control, enable the development of sophisticated measurement applications. The highly extensible .NET class libraries in Measurement Studio deliver unparalleled flexibility to scientists and engineers.

Visual C++ Class Libraries

Measurement Studio delivers an interactive design approach for developing virtual instruments in Visual C++. Measurement Studio includes Visual C++ classes to interface with hardware, provide data analysis, and transfer data across the Internet, as well as custom-wrapped ActiveX controls 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 Tools

The Measurement Studio tools integrate into Visual Studio .NET so you can use them exactly as you would native Microsoft tools. Measurement Studio dramatically reduces application development time with project templates, wizards, and hardware API assistants. Use a Measurement Studio project template to create a project according to your specifications, with an included code template and the measurement tools you need to design your measurement application. Use the Measurement Studio hardware API assistants—NI Instrument I/O Assistant and NI DAQ Assistant—to interactively define reusable measurement and generation tasks and automatically generate code for your project.

Related Software Packages

Measurement Studio offers additional packages for targeted applications. Visit 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, LabWindows™/CVI™, and Measurement Studio for developing test routines and TestStand for managing test execution, test sequencing, collecting data, and generating reports. The Test Edition also includes a comprehensive set of LabVIEW, LabWindows/CVI, 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. For more information about TestStand, visit ni.com/teststand.

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 an 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 moves, the preview window changes to display trajectory and velocity. Pan, zoom, and rotate the preview window to inspect moves in three dimensions.

Automation Symbols Toolset

The Automation Symbols Toolset contains a collection of ActiveX user interface 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. This toolset, which includes an ActiveX control, implements the standard PID algorithm and 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

The Measurement Studio Evaluation package contains the Measurement Studio evaluation software. Any applications you build with the Measurement Studio Evaluation package have a five minute run-time timeout.

System Requirements

To install Measurement Studio Evaluation package, 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 development 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.

Installation Instructions

Complete the following steps to install the Measurement Studio Evaluation package:

1. Insert the Measurement Studio Evaluation 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.

9. From the feature tree, select the Device Drivers components you want to install. To change a driver's 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.

Using Measurement Studio Help

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 so that you can quickly access help as you develop. 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*.

Context-Sensitive Help

As you develop with Measurement Studio, you can use the context-sensitive help to quickly access function reference. You can use the following types of context-sensitive help:

- F1 help—As you write code, put your cursor in a function name and press <F1> to access the function reference for that function.
- Dynamic Help window—As you write code, the Visual Studio .NET Dynamic Help window updates to display links to relevant topics about the Measurement Studio class libraries. To view the Dynamic Help window in Visual Studio .NET, select **Help»Dynamic Help**.
- What's This? help—When you use the property pages for ActiveX controls in the Measurement Studio Visual C++ class libraries, right-click properties in the property pages and select **What's This?** to learn more about the properties.

What is Measurement Studio?

Measurement Studio extends the .NET Framework and MFC and enhances Visual Studio .NET. The Measurement Studio class libraries enhance Visual C#, Visual Basic .NET, and Visual C++ by seamlessly integrating measurement and automation tools, rich, object-oriented measurement hardware interfaces, advanced analysis class libraries, scientific user interface controls, live measurement data networking, wizards, interactive code designers, and highly extensible .NET and C++ classes.



Note This evaluation package includes Measurement Studio tools only for Visual C#, Visual Basic .NET, and Visual C++ .NET. For Visual Basic 6.0 and Visual C++ 6.0 support, you can purchase the Measurement Studio Professional Edition or Measurement Studio Enterprise Edition. Visit ni.com/mstudio for details.

Why Should I Use Measurement Studio?

Measurement Studio provides enhancements designed to optimize the Visual Studio .NET environment—including Visual C#, Visual Basic .NET, and Visual C++—for measurement and automation applications and greatly reduces development time for scientists and engineers. Measurement Studio includes the following key enhancements:

- Intuitive measurement hardware APIs and assistants
- Powerful, flexible measurement classes
- Native .NET user interface controls
- Sophisticated analysis libraries
- Industry-leading software expertise

Intuitive Measurement Hardware APIs and Assistants

Intuitive measurement hardware APIs speed the development of I/O communication applications by abstracting low-level hardware implementation details to a higher level for rapid application development of DAQ, GPIB, VISA, and serial applications through native .NET and C++ interfaces. In addition to the interfaces, Measurement Studio also provides innovative, code-generating hardware API assistants that you can use to create measurement applications.

Extensible Measurement Classes

Extensible measurement classes enable flexible functionality for your specific application needs, such as defining a custom plot style for the Measurement Studio graph or creating a VISA passport for your proprietary bus. The Measurement Studio .NET and Visual C++ classes are architected to expose base class functionality that simplifies customization.

Native .NET User Interface Controls

Native .NET user interface controls, including 2D graphs, legends, switches, and LEDs, significantly reduce development time by providing professional, pre-built controls designed to display measurement data. In addition to .NET user interface controls, Measurement Studio presents user interface controls seamlessly in Visual C++, as multithread-safe MFC class controls.

Advanced Analysis Libraries

The Measurement Studio advanced analysis libraries power sophisticated application development. Incorporating proven analysis functionality into your application enables enhanced measurement results with decision-making capabilities, signal generation, advanced signal processing, and specific data analysis.

Industry-Leading Software Expertise

With more than 15 years of industry-leading software expertise, NI assures proven measurement and automation application development. Since the inception of Visual Basic, Measurement Studio has seamlessly integrated into Microsoft development environments by providing native measurement and automation add-ins.

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 develops native measurement tools for Visual Studio—Visual C#, Visual Basic .NET, and Visual C++—so you can benefit from measurement expertise without learning a new language. To integrate most efficiently into Visual Studio .NET, 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 languages and Visual Studio. As a developer, you have less to learn and can start using Measurement Studio tools quickly.

Developing Your Application with Measurement Studio

Measurement Studio provides class libraries and development tools to get your applications started quickly, which allows you to focus on your measurement task instead of low-level programming details. Measurement Studio dramatically reduces application development time through project templates, interactive wizards, code-generating hardware API assistants, and fully integrated help topics.

Measurement Studio delivers I/O, analysis, presentation, and other measurement-specific libraries to help you build your measurement and automation system through intuitive interfaces, including .NET class libraries for Visual Basic .NET and Visual C#, MFC-based classes for Visual C++, and ActiveX interfaces for Visual Basic.

In this chapter, the .NET and Visual C++ class libraries are discussed separately so that you can learn specifically about your preferred language.



Note For a complete list of all Measurement Studio class libraries, refer to the *Measurement Studio .NET Class Hierarchy Chart* and *Measurement Studio Visual C++ Class Hierarchy Chart*, which are included with this evaluation package. For function reference and in-depth documentation on developing with Measurement Studio, refer to the *NI Measurement Studio Help*.

Getting Started

This section describes how you can use the Measurement Studio development tools and class libraries to quickly create a measurement and automation application.

Creating a New Measurement Studio Project

When you use Measurement Studio to develop applications in Visual Studio .NET, take advantage of Measurement Studio project templates to quickly create an application in Visual C#, Visual Basic .NET, or Visual C++, as shown in Figure 4-1. When you use a Measurement Studio project template, you specify which Measurement Studio libraries to use, and references are automatically added to the project. You also can create projects that include DAQmx or instrumentation tasks.

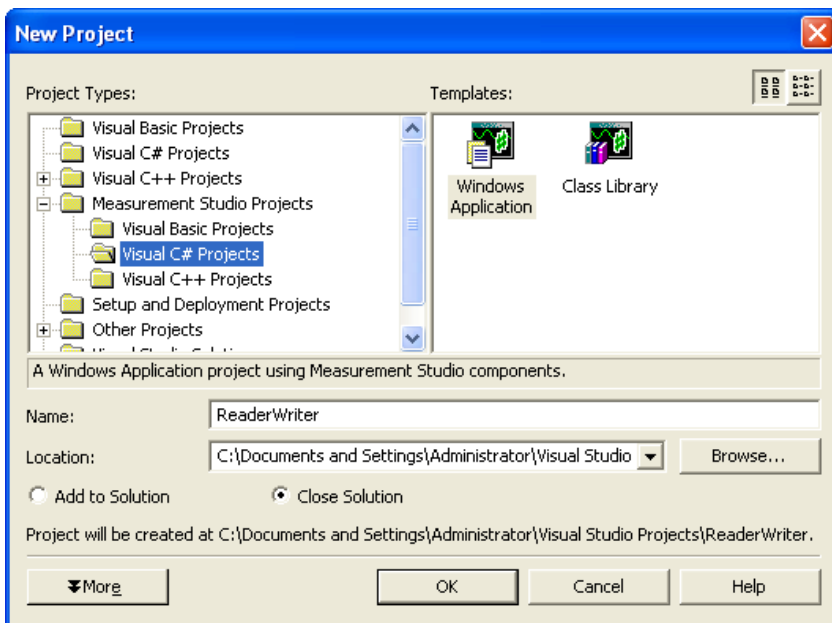


Figure 4-1. Create a Measurement Studio Project

Refer to the *Creating a New Measurement Studio Project* section of the *NI Measurement Studio Help* for more information.

Creating a Measurement Studio DAQ Project

To create a Measurement Studio DAQ project, take advantage of the DAQ Assistant. The DAQ Assistant is fully integrated into Visual Studio .NET as a code designer, as shown in Figure 4-2. With the DAQ Assistant, you can quickly and easily create and configure measurement and generation tasks. After you create a DAQmx task, the DAQ Assistant automatically generates Visual C#, Visual Basic .NET, or Visual C++ code for you.

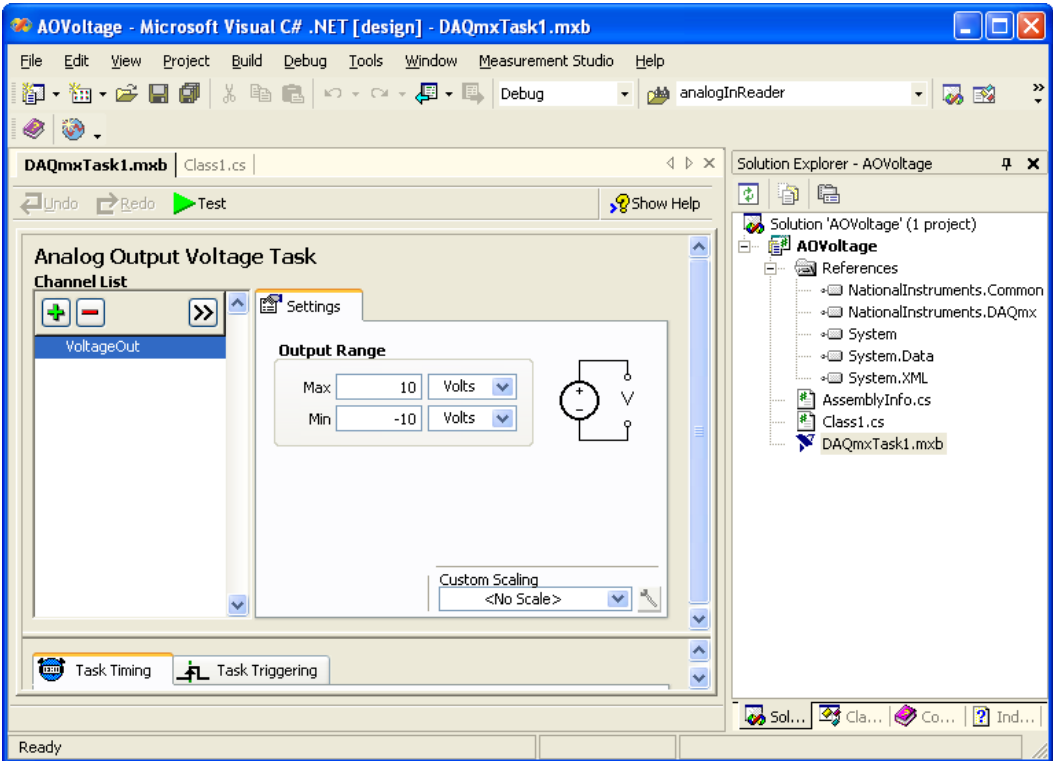


Figure 4-2. Use the DAQ Assistant in Visual Studio .NET

Refer to the *Creating a Measurement Studio DAQ Application* section of the *NI Measurement Studio Help* for more information.

Creating a Measurement Studio 488.2 or VISA Project

To create a Measurement Studio 488.2 or VISA project, you can use the NI Instrument I/O Assistant. The Instrument I/O Assistant is fully integrated into Visual Studio .NET as a code designer. With the Instrument I/O Assistant, you can quickly and easily create and configure instrumentation tasks. After you create an instrumentation task, the Instrument I/O Assistant automatically generates Visual C#, Visual Basic .NET, or Visual C++ code in your project. You can use this code to communicate with and control devices such as serial, Ethernet, or GPIB instruments. Refer to the *Creating a Measurement Studio 488.2 or VISA Application* section of the *NI Measurement Studio Help* for more information.

Adding or Removing Measurement Studio Class Libraries

If you need to add a reference to a Measurement Studio .NET or Visual C++ class library in an existing project, you can use the Measurement Studio Add/Remove Class Libraries wizard. This wizard provides an interface, shown in Figure 4-3, that you can use to select the class libraries you want to add to your project.

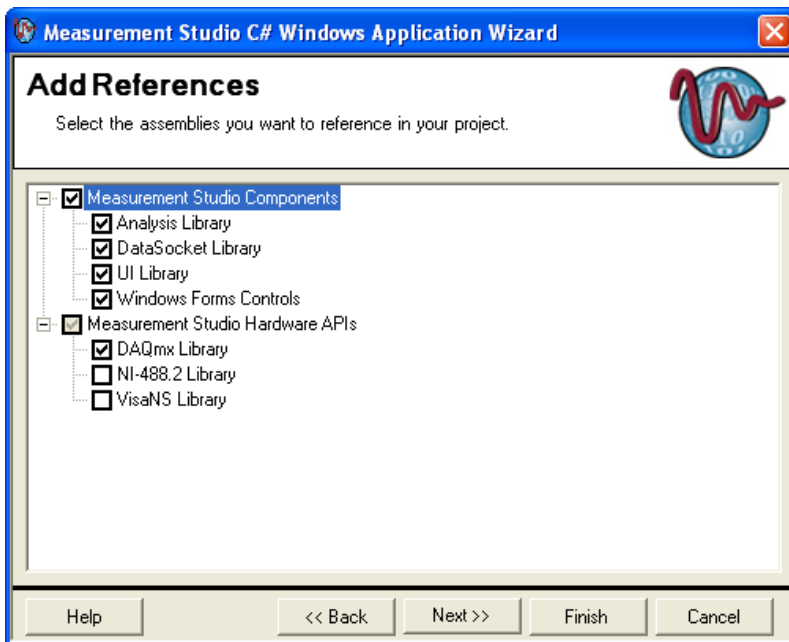


Figure 4-3. Add References to Measurement Studio Class Libraries

You also can use the Add/Remove Class Libraries wizard to remove references to Measurement Studio class libraries from a project. For more information, refer to the *Adding or Removing Measurement Studio Class Libraries* section of the *NI Measurement Studio Help*.

Using the Measurement Studio Tabs in the Visual Studio .NET Toolbox

If the project includes a form, the Measurement Studio .NET Tools tab and the Measurement Studio Visual C++ Tools tab are displayed in the Visual Studio .NET Toolbox. These tabs, shown in Figure 4-4, include icons for Measurement Studio .NET controls and Visual C++ ActiveX controls, respectively. When you select a Measurement Studio control from the tab and place it on the form, Visual Studio .NET automatically adds references to the project.

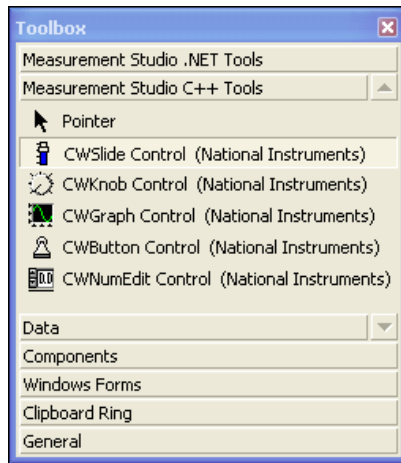


Figure 4-4. Use the Measurement Studio Toolbox Tabs to Add Controls to a Form

For more information about the Measurement Studio user interface controls, refer to the *Using the Measurement Studio .NET Class Libraries* section or *Using the Measurement Studio Visual C++ 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. To simplify this task, you can use the Measurement Studio Parameter Assistant to discover and automatically insert valid parameter values into your code. The Parameter Assistant, shown in Figure 4-5, finds parameter values for DataSocket, DAQ, NI-488.2, and VISA methods.

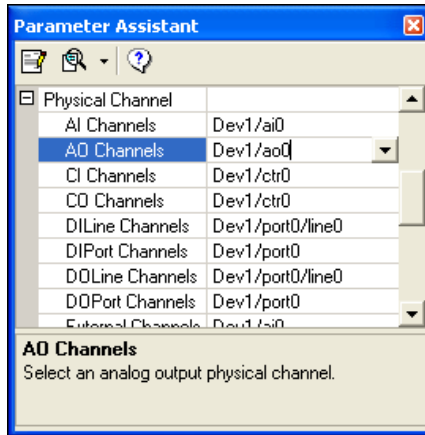


Figure 4-5. Select Parameter Values with the Parameter Assistant

For more information, refer to the *Selecting a Measurement Studio Parameter Value* section of the *NI Measurement Studio Help*.

Measurement Studio .NET Class Libraries

The Measurement Studio .NET class libraries are designed specifically for engineers and scientists who build virtual instrumentation systems with Visual C# or Visual Basic .NET in Visual Studio .NET. With integrated .NET libraries for acquiring, analyzing, and displaying data, Measurement Studio includes everything you need for building advanced measurement and automation applications. Measurement Studio provides the following .NET libraries and controls:

- NI-DAQmx class library
- NI-488.2 class library
- NI-VISA class library
- Analysis class library
- Waveform graph and scatter graph controls
- Switch and LED controls
- Legend control
- DataSocket class library

For more information about Measurement Studio .NET class libraries and controls, refer to the *Using the Measurement Studio .NET Class Libraries* section of the *NI Measurement Studio Help*.

Acquire

Data Acquisition

With the Measurement Studio NI-DAQmx .NET class library, you can easily communicate with and control NI data acquisition (DAQ) devices. You can 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

To quickly and easily create a DAQ application, use the DAQ Assistant. You do not need to perform low-level programming to set up your acquisition routines or to transfer buffers from your board to your computer—the DAQ Assistant handles the details for you. For more information about the DAQ Assistant, refer to the [Creating a Measurement Studio DAQ Project](#) section.

Instrument Control

Whatever you use to acquire your data—a GPIB or serial instrument, a plug-in data acquisition card, PXI hardware, or an NI computer-based instrument—Measurement Studio provides the high-level interface native to your development environment. For instrument control, Measurement Studio includes the NI-488.2 and NI-VISA .NET class libraries.



Tip To quickly and easily create a VISA or 488.2 application, use the Instrument I/O Assistant. For more information about the Instrument I/O Assistant, refer to the [Creating a Measurement Studio 488.2 or VISA Project](#) section.

NI-488.2

The NI-488.2 .NET class library includes a set of classes that communicates with GPIB instruments, controls GPIB boards, and acquires GPIB status information. You can use this library to quickly create programs that interface with a device that is using GPIB and/or programs that interface with GPIB directly.

NI-VISA

The NI-VISA .NET class library includes a set of classes that provides a rich, object-oriented interface to the NI-VISA driver. You can use this library to quickly create bus-independent and/or bus-specific instrument control applications. You can use this library to 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



Note National Instruments has designed the VisaNS API as a custom, optimized .NET interface to NI-VISA.

Analyze

Measurement Studio delivers the Analysis .NET class library 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.

Present

The Measurement Studio .NET user interface controls deliver scientific and engineering presentation capabilities in an intuitive and extensible fashion. You can easily configure the user interface controls programmatically or through the Visual Studio .NET Properties window in the Windows Forms Designer. The controls also provide various collection editors to make it easier to configure collections of subobjects.

Scatter Graph and Waveform Graph Controls

You can use the scatter graph and waveform graph controls 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. Both graph

controls include many properties that you can use to configure the appearance and behavior of the graph. For example, you can easily configure properties such as the caption text, caption background color, and color of the graph frame.

Switch and LED Controls

In addition to the graph controls, Measurement Studio provides switch and LED controls that you can use on a Windows Forms user interface. You typically use a switch control to receive Boolean input in an application user interface. The LED control represents a light-emitting diode. You typically use an LED to indicate a Boolean value on an application user interface. Both controls include many properties you can use to configure the appearance and behavior of the control.

Legend Control

You can use the legend control to display symbols and descriptions for a specific set of elements of another object, such as plots or cursors on a graph. When you associate the legend control with another object, any changes you make to that object are 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.

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.

Use the simple DataSocket API to communicate with the following types of servers:

- DataSocket Servers (`dstp:`)
- Web servers (`http:`)
- File transfer protocol servers (`ftp:`)
- File systems (`file:`)
- OLE for Process Control servers (`opc:`)



Tip The Measurement Studio user interface controls include built-in DataSocket connectivity. You can create a simple, yet powerful Visual C# or Visual Basic .NET program to visualize measurements or publish them from your desktop.

Measurement Studio Visual C++ Class Libraries

Measurement Studio delivers robust I/O, analysis, presentation, and other measurement-specific libraries and controls to help you build your measurement and automation system through object-oriented classes in Visual C++. Measurement Studio provides the following Visual C++ libraries and controls:

- NI-DAQmx class library
- NI-488.2 class library
- NI-VISA class library
- FieldPoint class library
- LabVIEW Real-Time interface
- Instrument driver class libraries
- Analysis class library
- 3D and 2D graph controls
- Button, knob, slide, and numeric edit controls
- Microsoft Word and Excel interfaces
- DataSocket class library

For more information about Measurement Studio Visual C++ class libraries, refer to the *Using the Measurement Studio Visual C++ Class Libraries* section of the *NI Measurement Studio Help*.

Acquire

Data Acquisition

For data acquisition, Measurement Studio provides the NI-DAQmx, FieldPoint, and LabVIEW Real-Time Interface class libraries.

FieldPoint

You can use the Measurement Studio FieldPoint Visual C++ class library to design Visual C++ applications that communicate with National Instruments FieldPoint modules. The FieldPoint class library provides programmatic access to all FieldPoint configuration options over Ethernet. For more information about FieldPoint, visit ni.com/fieldpoint.

LabVIEW Real-Time Interface

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

NI-DAQmx

With the Measurement Studio NI-DAQmx Visual C++ class library, you can easily communicate with and control NI data acquisition (DAQ) devices. You can 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

To quickly and easily create a DAQ application, use the DAQ Assistant. You do not need to perform low-level programming to set up your acquisition routines or to transfer buffers from your board to your computer—the DAQ Assistant handles the details for you. For more information about the DAQ Assistant, refer to the [Creating a Measurement Studio DAQ Project](#) section.

Instrument Control

Whatever you use to acquire your data—a GPIB or serial instrument, a plug-in data acquisition card, PXI hardware, or an NI computer-based instrument—Measurement Studio provides the high-level interface native to your development environment. For instrument control, Measurement Studio includes the IVI instrument driver, NI-488.2, and NI-VISA Visual C++ class libraries.

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.

IVI instrument drivers are divided into two main categories—class drivers and specific drivers. Measurement Studio includes interfaces for the following class drivers:

- IVI DC power supply
- IVI DMM
- IVI arbitrary waveform generator
- IVI oscilloscope
- IVI switch

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.

NI-488.2

The NI-488.2 Visual C++ class library includes a set of classes that communicates with GPIB instruments, controls GPIB boards, and acquires GPIB status information. You can use this library to quickly create programs that interface with a device that is using GPIB and/or programs that interface with GPIB directly.

NI-VISA

The NI-VISA Visual C++ class library includes a set of classes that provides a rich, object-oriented interface to the NI-VISA driver. You can use this library to quickly create bus-independent and/or bus-specific instrument control applications. Also, you can use this library to 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

Analyze

Measurement Studio delivers the Analysis Visual C++ class library 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.

Present

User Interface Controls

Measurement Studio delivers ready-to-use, measurement-specific user interface controls 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. 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, and dials
- Binary switches and LEDs
- Slides, tanks, and thermometers
- Real-time 2D and 3D graphs

Microsoft Word and Excel Interfaces

Measurement Studio provides the Microsoft Word and Excel interfaces so that you can perform offline processing of the measurement and automation data you acquire and analyze with other Measurement Studio Visual C++ class libraries. Use the Microsoft Word Interface to automatically create Word documents from within measurement and automation applications. Use the Microsoft Excel Interface to automatically create Excel spreadsheets and charts from within measurement and automation applications.

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.

Use the simple DataSocket API to communicate with the following types of servers:

- DataSocket Servers (`dstp:`)
- Web servers (`http:`)
- File transfer protocol servers (`ftp:`)
- File systems (`file:`)
- OLE for Process Control servers (`opc:`)

Distributing a Measurement Studio Application

The release version of Measurement Studio includes merge modules that you can use to build MSI-based installers for applications that use Measurement Studio class libraries. For more information about distributing applications developed with Measurement Studio, refer to the *Distributing Measurement Studio Applications* section of the *NI Measurement Studio Help*.



Note You cannot distribute applications that you build with the Measurement Studio Evaluation package.

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 measurement and automation development in the language of their choice.

Related Documentation

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

- *Measurement Studio User Manual*—Includes information about Measurement Studio class libraries and tools. This manual is available as a PDF from the Measurement Studio Evaluation CD.
- *NI Measurement Studio Help*—Includes complete function reference and in-depth documentation on developing with Measurement Studio.
- *Measurement Studio .NET Class Hierarchy Chart* and *Measurement Studio Visual C++ Class Hierarchy Chart*—Provide overviews of class relationships within class libraries.
- Measurement Studio Web site—Provides up-to-date information about Measurement Studio. Visit ni.com/mstudio.



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.

Measurement Studio Examples

Measurement Studio installs example programs for Visual C#, Visual Basic .NET, and Visual C++. Measurement Studio installs examples to the following paths:

- Visual C# or Visual Basic .NET—Program Files\National Instruments\MeasurementStudio70\DotNET\Examples
- Visual C++—Program Files\National Instruments\MeasurementStudio70\VCNET\Examples

You also can visit NI Developer Zone at 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.

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	<ol style="list-style-type: none"> 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 <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 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	<ol style="list-style-type: none"> 1. ActiveX control. See ActiveX control. 2. Object for entering or manipulating data on a user interface. Compare with <i>indicator</i>.

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. <i>See also</i> 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 <code>.exe</code> 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; 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 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	<ol style="list-style-type: none">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 set of channels and the channel configurations, timing, triggering, and other details that define a measurement or generation that 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 *VXIplug&play* Systems Alliance.

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