RELEASE NOTES

Measurement Studio

These release notes introduce Measurement Studio 8.1. Refer to this document for installation requirements, deployment requirements, installation instructions, information about new features and functionality, and resources in Measurement Studio.

These release notes are an extension of the *Measurement Studio User Manual*, which has not been updated for Measurement Studio 8.1. Select **Start»All Programs»National Instruments»<Measurement Studio>»Measurement Studio User Manual** to access the *Measurement Studio 8.0.1 User Manual*.

Installation Requirements

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

- Microsoft Windows 2000/XP/XP x64[1]
- Microsoft .NET Framework 1.1 for Visual Studio .NET 2003 or Microsoft .NET Framework 2.0 for Visual Studio 2005 (required only for the Measurement Studio .NET class libraries)
- Standard, Professional, Enterprise Developer, Enterprise Architect, or Academic edition of Microsoft Visual Studio .NET 2003 (required to use the Measurement Studio integrated tools); Standard, Professional, or Team System edition of Microsoft Visual Studio 2005 (required to use the Measurement Studio integrated tools); or Visual C#, Visual Basic .NET, or Visual C++ Express Editions of Microsoft Visual Studio 2005
- Intel Pentium II class processor, 733 MHz or higher
- Video display—800 × 600, 256 colors (16-bit color recommended for user interface controls)
- Minimum of 256 MB of RAM (512 MB or higher recommended)

[1] You cannot use Measurement Studio class libraries in 64-bit applications. You can, however, use Measurement Studio class libraries in 32-bit applications and run those applications on XP x64, provided that all drivers you use in the applications support XP x64. Current NI driver support for XP x64 is limited.

NATIONAL INSTRUMENTS

- Minimum of 405 MB of free hard disk space for Visual Studio .NET 2003 support and minimum of 385 MB of free hard disk space for Visual Studio 2005 support
- Microsoft-compatible mouse
- Microsoft Internet Explorer 6.0 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 or Microsoft .NET Framework SDK 2.0.

Deployment Requirements

Installing the Current Version of Measurement Studio over Previous Versions of Measurement Studio

You can have only one version of Measurement Studio installed on a system for each version of Visual Studio or the .NET Framework installed on the system. For example, you can have Measurement Studio 8.0.1 for Visual Studio .NET 2003 installed on the same system as Measurement Studio 8.1 for Visual Studio 2005, but you cannot have Measurement Studio 8.0.1 for Visual Studio 2005 installed on the same system as Measurement Studio 8.1 for Visual Studio 2005.

If you install a never version of Measurement Studio on a machine that has a prior version of Measurement Studio installed, the newer version installer replaces the prior version functionality, including class libraries. However, the prior version assemblies remain in the global assembly cache (GAC); therefore, applications that reference the prior version continue to use the prior version .NET assemblies.[1]

The default directory for Measurement Studio 8.1 support for Visual Studio .NET 2003 (`Program Files\NationalInstruments\MeasurementStudioVS2003`) is different than the default directory for Measurement Studio 7.0 support for Visual Studio .NET 2003 (`Program Files\NationalInstruments\MeasurementStudio70`). If Measurement Studio 7.0 is installed on your machine when you install Measurement Studio 8.1, Measurement Studio 8.1 installs to the 7.0 directory. If you prefer to install Measurement Studio 8.1 to the default 8.1 directory, you must first uninstall all Measurement Studio class libraries, including class libraries installed with

[1] This does not apply to `NationalInstruments.Common.dll`. `NationalInstruments.Common.dll` uses a publisher policy file to redirect applications to always use the newest version of `NationalInstrumenst.Common.dll` installed on the system, for each version of the .NET Framework. `NationalInstruments.Common.dll` is backward compatible.

National Instruments driver software, such as NI-VISA, NI-488.2, and NI-DAQmx.

Installing Measurement Studio

Complete the following steps to install Measurement Studio. These steps describe a typical installation. Please carefully review all additional licensing and warning dialog boxes.

Note There are separate installers for Measurement Studio support for Visual Studio .NET 2003 and Measurement Studio support for Visual Studio 2005. Repeat the installation instructions to install support for both.

1. Insert the Measurement Studio CD into the CD drive. `autorun.exe` automatically starts. If it does not automatically start, double-click the `autorun.exe` icon.
2. Click **Install Measurement Studio for Visual Studio .NET 2003** or click **Install Measurement Studio for Visual Studio 2005**.
3. Enter the serial number. You can find your serial number on the Certificate of Ownership card that you received with your Measurement Studio software. Click **Next**.
4. Review the information in the Product Information dialog box and click **Next**.
5. Click **Next** to install all NI software to the default installation directory, or click **Browse** to select a different installation directory. Click **Next**.

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

6. From the feature tree, select the features you want to install. To change the Measurement Studio installation directory, select the first feature in the list and click **Browse**. You must install Measurement Studio to a local drive. If you install Measurement Studio support for more than one version of Visual Studio, install them to different directories. Click **Next**.
7. Review the license agreement and select **I accept the License Agreement(s)**. Click **Next**.
8. In the Installation Summary dialog box, review the features you selected. Click **Next**.

Note Step 9 starts the installation of Measurement Studio. Be aware that when the installer indicates that it is removing backup files, this is a normal operation. The installer may take several minutes to complete this step.

9. If prompted, insert the Device Drivers CD and select **Rescan Drive**. If not prompted, go to step 15 on this list.
10. 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**.
11. In the Product Information dialog box, carefully review important information about the features you are installing. Click **Next**.
12. Review the license agreement and select **I accept the License Agreement(s)**. Click **Next**.
13. In the Installation Summary dialog box, review the features you selected. Click **Next**.
14. Click the **Register** button to register Measurement Studio now, and click **Next** to complete the installation.
15. If prompted, click the appropriate restart option. If you did not install a component that requires a restart, you will not be prompted to restart.

What's New in Measurement Studio 8.1

Measurement Studio includes support for Visual Studio 2005, Visual Studio .NET 2003, and Visual Studio 6.0. Measurement Studio 8.1 includes a new class library, the Network Variable class library, and a new user interface control, the instrument strip control. Measurement Studio 8.1 includes incremental enhancements to existing Measurement Studio support for Visual Studio .NET 2003, including error bands for the waveform and scatter graph controls.

Note Measurement Studio support for Visual Studio 6.0 is now considered legacy. There have been no changes to this product since Measurement Studio 8.0.1. The label on this CD reads Measurement Studio 8.0.1 Support for Visual Studio 6.0.

Measurement Studio Support for Visual Studio .NET 2003 New Features

Measurement Studio 8.1 includes waveform, scatter, and complex graph enhancements for use with Visual Studio .NET 2003. These enhancements are also available in Measurement Studio support for Visual Studio 2005.

Note In Measurement Studio support for Visual Studio .NET 2003, several dialog boxes refer to Measurement Studio 8.0.1, even though the product is Measurement Studio 8.1. The following dialog boxes contain the outdated version number: Licenses, About NI Measurement Studio, Preferences, and the Project Conversion Wizard.

Waveform and Scatter Graph Enhancements: Error Bands

With the Measurement Studio waveform and scatter graph .NET controls, you can calculate and display error bands.

Figure 1 shows error bands in Windows Forms.

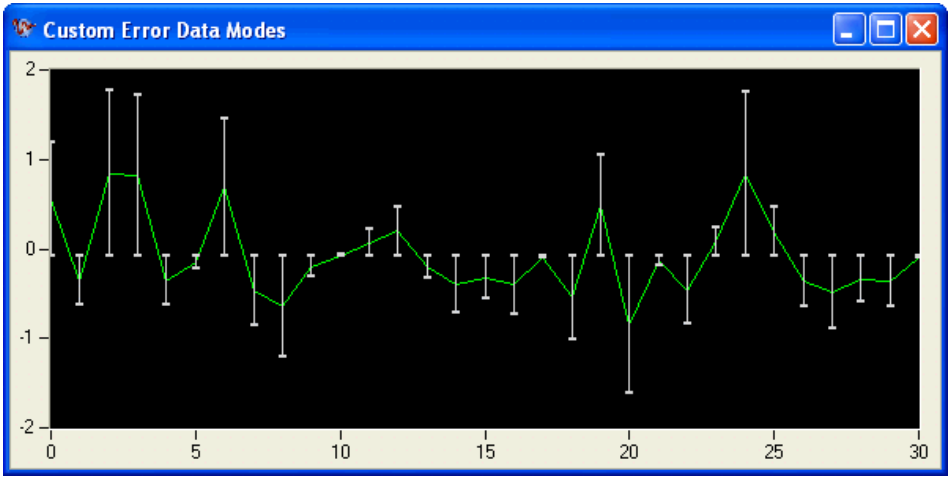

Figure 1. Windows Forms Error Bands

Figure 2 shows error bands in Web Forms.

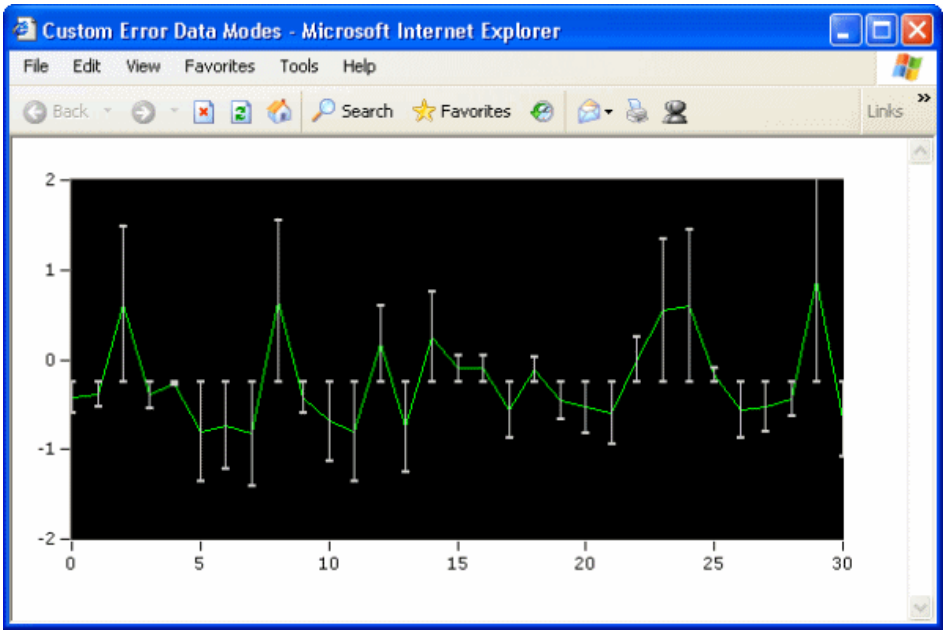

Figure 2. Web Forms Error Bands

With error bands, you can perform the following operations:

- Calculate error data with common modes, such as percentage and constant values.
- Create custom error data modes if your error data does not fit one of the common modes.
- Use `MapXErrorHighData`, `MapXErrorLowData`, `MapYErrorHighData`, and `MapYErrorLowData` methods to transform error data stored with the plot to an array of points in device coordinates using specificied bounds and the x-axis and y-axis ranges.
- Customize the appearance, such as the line style or point style, of the high error band and low error band. You can customize the high and low error bands independently.

For more information, refer to the *Creating a Custom Error Data Mode for the Measurement Studio Scatter and Waveform Graph .NET Controls* topic or the *Creating a Custom Plot for the Measurement Studio Scatter and Waveform Graph .NET Controls* topic in the *NI Measurement Studio Help*.

Complex Graph Enhancements: Cursors and Annotations

The complex graph includes cursors and annotations.

Cursors

You can use cursors to perform the following operations:

- Identify key points in plots and the plot area.
- Configure cursor snap modes to be fixed, floating, snap to nearest point, or snap to plot.
- Use cursor labels to display real, imaginary, magnitude, and phase data coordinates that the cursor crosshair points to, and customize the text font and colors of the label.
- Create custom point and line styles for cursors.

Figure 3 shows the Measurement Studio Windows Forms complex graph with cursors and annotations.

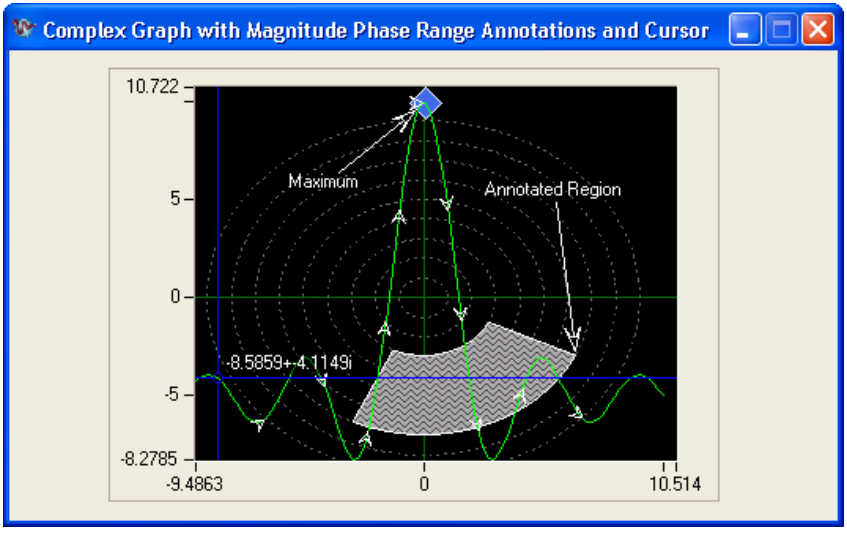

Figure 3. Windows Forms Complex Graph with Magnitude Circle, Magnitude Phase, and Complex Point Annotations and Cursors

Annotations

You can use annotations to perform the following operations:

- Configure a label for a point, a label for a range of real and imaginary values, a label for a range of magnitude values for a particular phase, or a label for a magnitude value.
- Configure text labels, arrows, and drawing shapes to annotate a point anywhere in the plot area of the graph.
- Configure range area, text labels, and arrows to annotate a range in the plot area of the graph.

For more information, refer to the *Key Measurement Studio Complex Graph .NET Control Features* topic in the *NI Measurement Studio Help*.

Measurement and Automation Explorer (MAX) Portable Configuration API

Use the MAX Configuration application programming interface (API) to copy saved configuration data to or from a system. You can use this API to:

- Package configuration data for deployment on other systems.
- Back up and restore your system configuration.

You must call the MAX Configuration API functions from a local system. You can use the API to copy data on remote systems to either a local system or another remote system, but you cannot run code on a remote system using the API.

Measurement Studio Support for Visual Studio 2005 New Features

The following features are available for use with Measurement Studio support for Visual Studio 2005.

Figure 4 shows the Measurement Studio Toolbox with new features, including the instrument control strip control, the network variable data source, and the network variable browser dialog items.

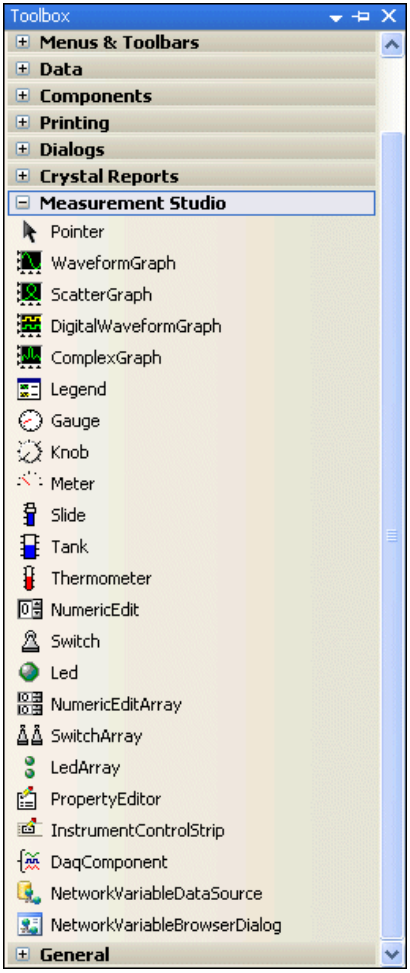

Figure 4. Measurement Studio group in Toolbox

Network Variable

The Measurement Studio Network Variable .NET class library includes three namespaces: `NationalInstruments.NetworkVariable`, `NationalInstruments.NetworkVariable.WindowsForms`, and `NationalInstruments.NetworkVariable.WebForms`. Use the Network Variable class library to transfer live measurement data between applications and servers over the network.

Note The Network Variable class library is intended to supersede the Measurement Studio DataSocket .NET class library. The Measurement Studio DataSocket .NET library is not available by default from the Toolbox in Visual Studio 2005. To add DataSocket to the Toolbox, right-click the Toolbox. Select **Choose Items**. In the Choose Toolbox Items dialog box, select **DataSocket**.

Use the features in the Network Variable class library to perform the following operations:

- Exchange different types of data between Measurement Studio, LabVIEW, LabWindows™/CVI™, and other applications that support NI-Publish Subscribe Protocol (psp:) and OLE for Process Control (opc:) servers. Exchanging data with OPC servers requires the LabVIEW Datalogging and Supervisory Control Module Run-Time System (LabVIEW DSC RST).

Note Network variables in Measurement Studio and LabWindows/CVI are equivalent to shared variables in LabVIEW. You can read to and write from Measurement Studio and LabWindows/CVI network variables with LabVIEW shared variables.

- Explicity create a variable using the stand-alone Variable Manager application. Refer to *Creating Shared or Network Variables with the Variable Manager* topic in the *Variable Manager Help* for information on how to create an explicit variable.
- Use Windows Forms and Web Forms data sources to expose Network Variable data items that you can bind to properties of a Windows Forms or a Web Forms control.
- Use the Network Variable Browser dialog box to quickly locate and select data items on other computers and servers. The Browser Dialog is included in the `NationalInstruments.NetworkVariable.WindowsForms` class.

Tip For more detailed information about the Network Variable class library, refer to the *Using the Meausrement Studio Network Variable .NET Library* section in the *NI Measurement Studio Help*.

Instrument Control Strip Control

Use the Measurement Studio instrument control strip control to display a set of Measurement Studio property editor controls through the `ToolStripPropertyEditor`. With the instrument control strip control, you can give your end users the ability to customize the user interfaces you develop. Use the instrument control strip control to add many property editor controls to your user interface without taking up a great deal of space.

Figure 5 shows the Measurement Studio instrument control strip control (circled).

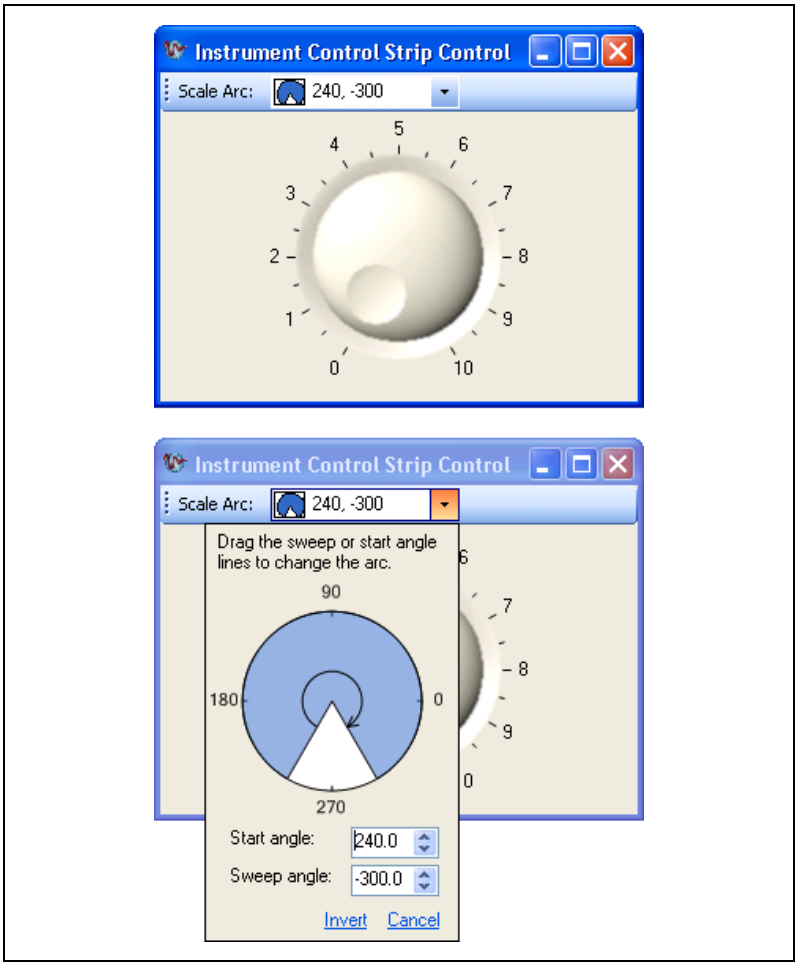

Figure 5. Instrument Control Strip Control

With the instrument control strip control and the classes that interface with the control, you can perform the following operations:

- Use the instrument control strip control as a toolbar for editing property values of another control through the associated editors at run time.
- Edit multiple property values of controls with one instrument control strip control.
- Add other types of controls, such as the tool strip button or tool strip label control, to the instrument control strip control.
- Customize the appearance of the control.

Tip For more information, refer to the *Using the Instrument Control Strip Control* topic in the *NI Measurement Studio Help*.

High Resolution Timing

The Measurement Studio Common .NET class library now includes the `PrecisionWaveformTiming` class and the `PrecisionTimeSpan` and `PrecisionDateTime` structures. `PrecisionWaveformTiming` provides higher precision timing than the `WaveformTiming` class. `PrecisionWaveformTiming` uses `PrecisionTimeSpan` and `PrecisionDateTime` to store timing information, and you can use the `PrecisionTimeSpan` or `PrecisionDateTime` objects to represent time that is accurate to the nearest 2^{-64} second. You use `PrecisionWaveformTiming` when working with `AnalogWaveform` and `DigitalWaveform` classes.

Measurement Studio Resources

As you work with Measurement Studio, you might need to consult additional 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 environment. The *NI Measurement Studio Help* is fully integrated with the Visual Studio help. You must have Visual Studio installed to view the online help, and you must have the Microsoft .NET Framework SDK 1.1 or the Microsoft .NET Framework SDK 2.0 installed in order for links from Measurement Studio help topics to .NET Framework help topics to work. You can launch the *NI Measurement Studio Help* in the following ways:

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

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

- Getting Started information—Refer to the *Measurement Studio Core Overview* topic and the *Getting Started with the Measurement Studio Class Libraries* section in the *NI Measurement Studio Help* for an introduction to Measurement Studio and for walkthroughs that guide

you step-by-step in learning how to develop Measurement Studio applications.

- Examples—Measurement Studio installs examples to the following paths:
  - Visual Basic .NET or Visual C#—`Program Files\National Instruments\<MeasurementStudio>\DotNET\Examples`
  - Visual C++—`Program Files\National Instruments\<MeasurementStudio>\VCNET\Examples`
- NI Technical Support—Refer to the *Technical Support and Professional Services* topic in the *NI Measurement Studio Help* for more information.
- Measurement Studio Web site, `ni.com/mstudio`—Contains Measurement Studio news, support, downloads, white papers, product tutorials, and purchasing information.
- NI Developer Zone, `zone.ni.com`—Provides access to online example programs, tutorials, technical news, and a Measurement Studio Discussion Forum where you can participate in discussion forums for Visual Basic 6.0, Visual C++, and .NET Languages.
- *Measurement Studio .NET Class Hierarchy Chart* and *Measurement Studio Visual C++ Class Hierarchy Chart*—Provide overviews of class relationships within class libraries. Charts are included with all Measurement Studio packages and are posted online at `ni.com/manuals`.