

RELEASE NOTES

Measurement Studio™

These release notes supplement the *Measurement Studio User Manual*. Refer to this document for information about new features and functionality, specific updates to the documentation, and resources in Measurement Studio. These release notes include information about Measurement Studio support for Visual Studio 2008, Visual Studio 2005, Visual Studio .NET 2003, and Visual Studio 6.0.

For installation instructions, installation requirements, deployment information, a list of fixed bugs, and known issues, refer to the *Measurement Studio Readme*. There is a different *Measurement Studio Readme* for each supported version of Visual Studio. The *Measurement Studio Readme* files are available in the root folder of the installation CD and are linked from the Autorun menu. After installing Measurement Studio 8.5, select **Start»All Programs»National Instruments»<Measurement Studio>¹»Readme** to access the *Measurement Studio 8.5 Readme*.

For a complete introduction to Measurement Studio and to learn about Measurement Studio concepts, controls, and features, refer to the *Measurement Studio User Manual*. Select **Start»All Programs»National Instruments»<Measurement Studio>»Measurement Studio User Manual** to access the *Measurement Studio User Manual*.

¹ There are separate Start menu items for each version of Visual Studio support.

Contents

What's New in Measurement Studio 8.5.....	3
Measurement Studio .NET Support for Visual Studio 2008.....	3
ASP.NET AJAX Compatibility.....	4
AutoRefresh Callback Improvements	4
Explicitly Creating Network Variables	4
Browsing Network Variables	4
Network Variable Buffer Flushing.....	5
Memory-Optimized Analog Waveform Read Methods.....	5
Visualization Support for ComplexWaveform in the WaveformGraph	6
.NET Analysis Library Code Snippets	6
Additional New Features	6
Limited Update to Visual Studio .NET 2003 Support	8
Visual Studio 6.0 Support.....	8
Documentation Updates	8
Learning Measurement Studio.....	8

What's New in Measurement Studio 8.5

New features in Measurement Studio 8.5 support for Visual Studio 2005 and Visual Studio 2008 include:

- Measurement Studio .NET support for Visual Studio 2008
- ASP.NET AJAX compatibility, available in Visual Studio 2008 only
- AutoRefresh callback improvements
- Explicitly creating network variables
- Browsing network variables
- Network variable buffer flushing
- Memory-optimized analog waveform read methods
- Visualization support for ComplexWaveform in the WaveformGraph
- .NET Analysis library code snippets
- Additional new features
- Limited update to Visual Studio .NET 2003 support
- Visual Studio 6.0 support

Measurement Studio .NET Support for Visual Studio 2008

Measurement Studio 8.5 includes .NET support for Visual Studio 2008, including a complete set of .NET class libraries and tools that integrate into Visual Studio 2008. To use these new libraries, either create new projects in Visual Studio 2008 or use Visual Studio 2008 to upgrade your existing Visual Studio 2005 projects that use Measurement Studio support for Visual Studio 2005 libraries. The Measurement Studio Visual Studio 2008 class libraries have different versions and different public key tokens than the corresponding Measurement Studio Visual Studio 2005 class libraries. The version numbers of Visual Studio 2008 support assemblies begin with 8.5.35; the version numbers of Visual Studio 2005 support assemblies begin with 8.5.20. If you use the public key token to verify the identity of Measurement Studio class libraries, you will need to modify your code or system configuration to accommodate the new public key token.



Note Measurement Studio 8.5 support for Visual Studio 2008 does not include support for Visual C++ class libraries.

For more information, refer to *Using Measurement Studio Assemblies for Visual Studio 2005 in Visual Studio 2008* in the *NI Measurement Studio Help*.

ASP.NET AJAX Compatibility

Measurement Studio Visual Studio 2008 Web Forms controls have been updated to integrate seamlessly with Microsoft ASP.NET AJAX controls. With this update, you can quickly create Web pages that include a rich user experience with responsive and familiar user interface elements. ASP.NET AJAX provides client-script libraries that incorporate cross-browser JavaScript and dynamic HTML (DHTML) technologies and integrates them with the ASP.NET server-based development platform. You can use ASP.NET AJAX server controls with Measurement Studio Web Forms controls to improve the user experience and the efficiency of your Web applications.

For more information, refer to *Measurement Studio AutoRefresh vs. ASP.NET AJAX UpdatePanel and Timer* in the *NI Measurement Studio Help*.

AutoRefresh Callback Improvements

This new feature provides a mechanism for updating the `Enabled` and `Interval` properties for `AutoRefresh` from within the `Refresh` callback, allowing you to turn off `AutoRefresh` or change the `Interval` during an asynchronous HTTP request without causing a postback. Previously, the JavaScript timer on the client did not get updated during an asynchronous HTTP request, and only the controls in the `AutoRefresh` collection were updated.

Explicitly Creating Network Variables

In addition to explicitly creating network variables with the `Variable Manager`, now you also can explicitly create network variables with the following classes: `ServerProcess`, `ServerProcessInfo`, `ServerVariable`, and `ServerVariableInfo`. You can use `ServerProcess` to perform a process-related operation; `ServerProcessInfo` and `ServerVariableInfo` to periodically monitor a network variable process or variable, respectively; and `ServerVariable` to perform a single variable-related operation.

For more information on using these classes to explicitly create and configure network variables, refer to *Explicitly Creating Network Variables* in the *NI Measurement Studio Help*.

Browsing Network Variables

The network variable API includes the `Browser` class and the network variable configuration classes, `ServerProcess`, `ServerProcessInfo`, `ServerVariable`, and `ServerVariableInfo`. You can use these classes to discover network variables and processes.

You can use the `Browser` class for the following tasks:

- Discovering processes and variables on local and remote machines
- Finding any item that exists in the network variable engine
- Accessing implicitly created network variables
- Determining the network variable type

You can use the `ServerProcess`, `ServerProcessInfo`, `ServerVariable`, and `ServerVariableInfo` classes for the following browse tasks:

- Accessing stopped processes
- Accessing and modifying variable configurations

For more information on browsing network variables, refer to *Comparing the Network Variable Browser Class and the Configuration Classes* in the *NI Measurement Studio Help*.

Network Variable Buffer Flushing

Measurement Studio 8.5 gives you advanced control over when the buffer of the network variable API is flushed, allowing you to optimize the performance of the APIs for your particular application.

All network variable readers and writers maintain an internal buffer for each connection. All writer classes maintain an internal buffer to protect against data loss. The buffer automatically flushes every 10 milliseconds or when the buffer becomes full, and you can use the `FlushAllConnections` method to manually flush the buffer. All network variable reader classes, `BufferedSubscriber<TValue>`, `NetworkVariableReader<TValue>`, and `NetworkVariableSubscriber<TValue>`, must flush the buffer in order to perform a read, since the readers must send a request to receive the value to the network variable engine.

For more information on buffer flushing behavior and how to maximize performance in context of the buffer, refer to *Buffer Flushing Behavior of the Network Variable Class Libraries* in the *NI Measurement Studio Help*.

Memory-Optimized Analog Waveform Read Methods

Memory-optimized analog waveform reads are a special case of analog waveform reads in the NI-DAQmx and NI-SCOPE .NET class libraries where the `NationalInstruments.AnalogWaveform<TData>` class re-uses the memory utilized by an existing `AnalogWaveform<TData>` instead of allocating memory for a new data array on every invocation. Memory is reallocated if the number of samples requested is greater than

the current size of the `AnalogWaveform<TData>`. This allows for much more efficient memory usage when performing multiple analog waveform reads. Memory-optimized analog waveform read methods were introduced in NI-DAQ 8.7 and NI-SCOPE .NET class libraries 1.1.

Visualization Support for ComplexWaveform in the WaveformGraph

You can use `PlotComplexWaveform<TData>` to plot complex waveforms on the Windows Forms and Web Forms waveform graphs. You can use the `ComplexWaveformPlotOptions.DisplayMode` property to specify whether the complex waveform displays samples or increments of time, and you can use the `ComplexWaveformPlotOptions.ScaleMode` property to configure whether the data as displayed in raw form or scaled. You can also customize the scaling behavior by deriving from `ScaleMode`. You can also use the `DataToPlot` property to configure the waveform graph to plot the real, imaginary, magnitude, or phase data.

.NET Analysis Library Code Snippets

Measurement Studio 8.5 includes analysis code snippets in the documentation that can be copied and pasted into an application and used without modification. The following classes include new example code snippets:

- `Anova1Way`
- `BesselLowpassFilter`
- `ExponentialFit`

For more information, refer to *Using the Measurement Studio Analysis .NET Library* in the *NI Measurement Studio Help*.

Additional New Features

The following new features were also introduced in Measurement Studio 8.5:

- **AutoScaleVisibleExact and AutoScaleVisibleLoose Axis Modes**—The `AxisMode` enumeration now includes `AutoScaleVisibleExact` and `AutoScaleVisibleLoose` values. You use these values to configure an axis to autoscale according to the range of data currently being displayed on the graph instead of according to the range of all data in the graph. These modes are available on the scatter, waveform, and complex graphs. Also, the autoscale is retained during zoom and pan operations.
- **Mutually Exclusive Functionality for Switch and LED Arrays**—With this update, you can now configure an array of Boolean controls, such as a `Switch` array or an `Led` array, so that only one element in the array can be true at a time. This feature allows you to

control or display the status of instruments that require only a single Boolean value to be true at any given time.

- **Enhanced Tick and Label Visibility Customizations in .NET Graph Controls**—With this update, you can hide the minimum and maximum ticks and labels for any axis on any .NET graph. Previously, the minimum and maximum values of an axis on the graph were always displayed with a tick and label even when they did not fall on a major division interval.
- **InteractionHistoryCount Property**—The new read-only `InteractionHistoryCount` property returns the number of items currently in the undo stack in the graph. With this new property, you can monitor the number of items remaining in the stack that holds interactive operations and add event handlers to use this information in your application.
- **Network Variable ConnectionFailed Event**—If a `NetworkVariableDataSource` attempts to connect to a network variable and fails, the `ConnectionFailed` event is raised. Handle this event to determine why connections are failing; use the `ReadFailed` or `WriteFailed` events to determine when reads or writes fail.
- **Additional Analysis .NET Class Library Methods**—The following method overloads have been added:
 - `Measurements.HarmonicAnalyzer(Double[], Int32, Double, Double, Double[], Double[], Double)`
 - `SignalProcessing.Differentiate(Double[], Double, Double[], Double[], DifferentiationMethod)`
 - `SignalProcessing.UnwrapPhase(Double[], PhaseUnit)`
 - `Statistics.RootMeanSquared(ComplexDouble[])`

The following methods have been added:

- `Measurements.CycleRmsAverage`
- `Measurements.PulseMeasurements`
- `Measurements.StateLevels`
- `Measurements.TransitionMeasurements`
- **Bug Fixes**—Measurement Studio 8.5 includes many fixes for previously reported bugs. Refer to the fixed bug chart in the *Measurement Studio 8.5 Readme* for more information. Select **Start»All Programs»National Instruments»«Measurement Studio»»Readme** to access the *Measurement Studio 8.5 Readme*.

Limited Update to Visual Studio .NET 2003 Support

Measurement Studio 8.5 support for Visual Studio .NET 2003 includes no new features. The only revisions to Visual Studio .NET 2003 support are bug fixes in the ActiveX controls in the `cwui.ocx` and `cw3dgrph.ocx` libraries. The bug fixes in these controls are documented in the Visual Studio 2005 support Readme. The *Measurement Studio Readme* for Visual Studio .NET 2003 support has not been updated to document these bug fixes.

Visual Studio 6.0 Support

The Measurement Studio 8.5 installation CDs include Visual Studio 6.0 support; however, this support is considered legacy, and no revisions were made for Visual Studio 6.0 in Measurement Studio 8.5. Updates to the Measurement Studio Support for Visual Studio 6.0 software are available on `ni.com` by selecting **Support»Drivers and Updates»Software»Measurement Studio**.

Documentation Updates

As Measurement Studio features and functionality expand with each release, the information in the *Measurement Studio User Manual* can become outdated. We have added this section to provide a list of changes or additions to the *Measurement Studio User Manual* content.

For Measurement Studio 8.5, the *Measurement Studio User Manual* contains the most recent documentation available.

Learning Measurement Studio

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 for Visual Studio .NET 2003, the Microsoft .NET Framework SDK 2.0 for Visual Studio 2005, or the Microsoft .NET Framework SDK 3.5 for Visual Studio 2008 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 also are 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. For an introduction to Measurement Studio resources, refer to the *Using the Measurement Studio Help* topic in the *NI Measurement Studio Help*.
- Examples—Measurement Studio installs examples organized by class library, depending on the component, the version of Visual Studio or the .NET Framework that the example supports, the version of Measurement Studio installed on the system, and the operating system. For more information on example locations, refer to the *Where To Find Examples* topic in the *NI Measurement Studio Help*.
- 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.
- Review the information from the Microsoft Web site on using Visual Studio.

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. 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.