

Measurement Studio

2019

Measurement Studio is an integrated suite of tools and class libraries designed to help developers create measurement and automation Windows Forms, Windows Presentation Foundation (WPF), and Web Forms applications using Microsoft .NET technologies.

Measurement Studio provides object-oriented measurement hardware interfaces, advanced analysis libraries, scientific user interface controls for Windows Forms, Windows Presentation Foundation (WPF), and Web Forms applications, measurement data networking libraries, project wizards, interactive code designers, and highly extensible .NET classes. You can use Measurement Studio to develop a complete measurement and automation application that includes data acquisition, analysis, and presentation functionalities.

The *Measurement Studio Getting Started Guide* introduces the concepts associated with the Measurement Studio class libraries and development tools. This document assumes that you have a general working knowledge of Microsoft Visual Studio and the .NET Framework for application development.

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

Refer to ni.com/mstudio for more information about the functionality and features included with each Measurement Studio edition.

Contents

Measurement Studio Activation.....	2
Concurrent Licensing.....	3
Driver Support.....	3
Deployment Requirements.....	3
Deploying an Application.....	3
Deploying 64-bit Applications.....	3
How to Create Applications with Measurement Studio Controls and Class Libraries	4
Measurement Studio Integrated Tools and Features.....	4
Measurement Studio Menu.....	4
Creating a Measurement Studio Project.....	6
Adding or Removing Measurement Studio .NET Class Libraries.....	6
Using the Instrument Driver Wizard.....	7

Measurement Studio .NET Class Libraries.....	7
64-bit Support in Measurement Studio .NET Assemblies.....	8
Analysis.....	8
Common.....	10
Data Transfer.....	10
Technical Data Management Streaming (TDMS).....	11
User Interface.....	12
Windows Presentation Foundation (WPF) Controls.....	12
Windows Forms Controls.....	13
ASP.NET Web Forms Controls.....	14
Learning Measurement Studio.....	15
Where to Go Next.....	15

Measurement Studio Activation

As you install Measurement Studio, you can either activate a license or initiate the evaluation period for the Enterprise license.

If you do not activate Measurement Studio during installation, you can activate after you install. To activate after you install:

- Use NI License Manager; or
- If you have Measurement Studio Visual Studio-integrated tools installed, open Visual Studio, and select **Measurement Studio»Activate Measurement Studio**.

After launching NI License Manager, you can use one of the following methods to activate Measurement Studio:

- If you received a USB memory stick for your installation media, then your serial number is printed on the box you received. Use your serial number to use NI License Manager to activate Measurement Studio. You can also use your serial number to activate Measurement Studio by selecting **Measurement Studio»Activate Measurement Studio** from within Visual Studio environment.
- If you downloaded Measurement Studio, then you can sign in to your NI account and activate Measurement Studio using your NI credentials.

The first time you use a Measurement Studio licensed component, a dialog box appears to confirm your license activation.



Note If you choose to activate Measurement Studio through a secure Internet connection while your system's date and time settings are incorrect, you may receive the NI Activation Wizard - Activation Error dialog box. To fix this issue, re-set your system's date and time settings to the correct date and select **Retry** or select **Other** in the NI Activation Wizard - Activation Error dialog box to use an alternate activation method.



Note Activating the TestStand Debug Deployment license will activate the Measurement Studio Debug license.

Concurrent Licensing

Measurement Studio supports concurrent licensing as part of a volume license agreement. If your software is a part of a Volume License Agreement (VLA), contact your VLA administrator for installation and activation instructions.

Driver Support

To use .NET class libraries that interface to NI device drivers, you must install the underlying device drivers in addition to the .NET class libraries. You can run the underlying device driver installers from the NI Device Drivers media included with Measurement Studio or you can download the device driver from ni.com. Refer to *National Instruments .NET Driver Support* for information on what drivers are supported for each version of the .NET Framework and the download location of the installer files.

Deployment Requirements

To deploy an application built using Measurement Studio .NET class libraries, .NET Framework 4.0 (minimum) is required. Measurement Studio applications can be deployed to any target machine that includes .NET Framework 4.0 or newer.



Note Measurement Studio support for all versions of Visual Studio uses the same `NationalInstruments.Common.dll` assembly, which is built against the .NET Framework version 4.0.

Deploying an Application

National Instruments recommends different deployment tools depending on the method you want to use to create a distribution. Use Measurement Studio Installer Builder to create and deploy a Measurement Studio application using MSI-based installer technology or use NI Package Builder to create and deploy a Measurement Studio application using package technology. For more information on deploying applications, refer to the *Measurement Studio Installer Builder Help*, which is accessible from within the Installer Builder application, or refer to the *NI Package Builder Help*, which is accessible from within the NI Package Builder application.

Deploying 64-bit Applications

All Measurement Studio class libraries that support 64-bit include both 32-bit and 64-bit deployment runtime installers. This is true regardless of whether the class library includes platform-specific (i.e., x86 or x64) or platform-agnostic (i.e., Any CPU) assemblies. Refer to *Deploying Windows Applications* in the installed *NI Measurement Studio Help* for more information on using 64-bit runtime installers.

How to Create Applications with Measurement Studio Controls and Class Libraries

The following online topics include overview information and step-by-step instructions on developing applications with Measurement Studio tools and features. Refer to the *Getting Started* section of the *Measurement Studio Help* for more information about the functionality of these tools and features.

- Creating a Measurement Studio Application with Windows Forms Controls and Analysis
- Creating a Measurement Studio Application with Windows Forms Controls and Network Variable

Measurement Studio Integrated Tools and Features

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

Refer to the *Creating Projects with Measurement Studio Core* section in the *NI Measurement Studio Help* for more information about the functionality of these tools and features.

Measurement Studio Menu

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

- **Measurement Studio Evaluation Home Page**—Launches the Try NI Measurement Studio for Visual Studio site on ni.com. This menu item appears only if you do not have a Measurement Studio license already activated.
- **Add/Remove .NET Class Libraries Wizard**—Launches the Add/Remove .NET Class Libraries wizard, which you can use to add or remove Measurement Studio class libraries or assemblies in existing Visual Basic .NET or Visual C# projects. This menu option is available only when you have a Visual C# or Visual Basic .NET project open.
- **Refresh Project License File**—Updates the `licenses.licx` file in a Measurement Studio project to the currently referenced Measurement Studio assemblies. The Refresh Project process works by going through the `licenses.licx` file line by line for the active project and removing each Measurement Studio licensed type that matches the Measurement Studio `PublicKeyToken`. After all Measurement Studio licensed types are removed from the `licenses.licx` file, the current Measurement Studio licensed types that are referenced by the project are added to the `licenses.licx` file. This ensures all Measurement Studio licensed types used by the project are added to the `licenses.licx` file. This menu option is available only when you have a Visual C# or Visual Basic .NET project open.

- **Refresh Web.Config File**—Updates the `Web.config` file in a Measurement Studio ASP.NET project to the currently referenced Measurement Studio assemblies. This menu option is available only when you have an ASP.NET website project open.
- **Update Measurement Studio Project References**—Updates any outdated Measurement Studio references to the latest version installed on the system. This menu option is available only when you have a Visual C# or Visual Basic .NET Windows Forms project open.
- **NI Tools**—Includes links to the following NI tools:
 - **Measurement & Automation Explorer (MAX)**—Use MAX to configure NI hardware; add new channels, interfaces, and tasks; execute system diagnostics; and view devices and instruments connected to the system. The MAX menu option is available only if you have MAX installed.
 - **NI I/O Trace**—Use NI I/O Trace to monitor, record, and display National Instruments API calls made by instrument connectivity applications. Use NI I/O Trace to quickly locate and analyze any erroneous National Instruments API calls that an application makes and verify that the communication with an instrument is correct. The NI I/O Trace menu item is available only if you have NI I/O Trace installed.
 - **Distributed System Manager**—Use Distributed System Manager to edit, create, and monitor network or shared variables. For more information, refer to the **NI Distributed System Manager Help**. In Distributed System Manager, select **Help»System Manager Overview** to access the help.
 - **Measurement Studio Installer Builder**—Launches the Installer Builder application, which helps you to create an installer for your application that includes NI libraries and products. Installer Builder ensures that National Instruments products and dependencies are included in your installer for proper run time behavior. For information on creating an installer using Installer Builder, refer to the *Measurement Studio Installer Builder Help*, accessible from within the Installer Builder application.
- **Measurement Studio Examples**—Launches the Measurement Studio example directory. You can use these example programs to help you learn and understand key concepts and to explore the functionality of Measurement Studio.
- **Measurement Studio Help**—Use the *Measurement Studio Help* to access detailed Measurement Studio help, including function reference, walkthroughs, and conceptual topic documentation on developing with Measurement Studio.
- **Measurement Studio Getting Started Guide**—Launches the Measurement Studio manual.
- **Measurement Studio Online Resources**—Includes links to the following NI websites:
 - **Measurement Studio Home Page**—Launches the Measurement Studio website at ni.com/mstudio, which contains Measurement Studio news, support, downloads, and evaluation software.
 - **Instrument Driver Network**—Launches the Instrument Driver Network (IDNet) website where you can find, download, or submit a driver to communicate with third-party instruments.
 - **Discussion Forums**—Launches the NI Discussion Forums website at forums.ni.com. With NI Discussion Forums, you can participate in discussion

forums and exchange code with measurement and automation developers around the world.

- **Technical Support**—Launches the NI Technical Support website at ni.com/support. These support 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, hardware schematics and conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, and a measurement glossary.
- **Check in NI VLM Licenses**—Checks in the Measurement Studio concurrent license that you have checked out from a VLM (Volume License Manager) server. If your license has already been checked in and you select this menu option, there is no change to your license status. This menu option is available only when you are using a VLM license.
- **Patents**—Provides information about National Instruments patents.
- **About Measurement Studio**—Launches the NI Measurement Studio About box, which provides version information.
- **Licenses**—Provides information about National Instruments licenses.
- **Check for Updates**—Launches NI Package Manager to check for updates to Measurement Studio components.
- **Activate Measurement Studio**—Launches the NI Activation Wizard, which guides you through activating Measurement Studio. This menu item appears only if you do not have a Measurement Studio license already activated.
- **Purchase Measurement Studio**—Launches the NI Measurement Studio Web page, where you can purchase a Measurement Studio license. This menu item appears only if you have not purchased a Measurement Studio license.

Creating a Measurement Studio Project

Measurement Studio includes class library and application templates that you can use to quickly create measurement applications with Visual Basic .NET and Visual C#. Use the Visual Studio New Project dialog box to access these templates and to create projects. You can create the following projects in Measurement Studio:

- Measurement Studio Visual Basic .NET project (WPF and Windows Forms)
- Measurement Studio Visual C# project (WPF and Windows Forms)
- Measurement Studio ASP.NET website



Tip For more information about using project templates to create a new Measurement Studio project, refer to the *Measurement Studio Projects* section in the *NI Measurement Studio Help*.

Adding or Removing Measurement Studio .NET Class Libraries

To add or remove Measurement Studio .NET class libraries from a project, use the Measurement Studio Add/Remove .NET Class Libraries wizard on the Measurement Studio menu. This wizard provides an interface that you can use to select the Measurement Studio .NET class libraries you want to add to or remove from a project.

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



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

Using the Instrument Driver Wizard

To use an IVI or VXI *plug&play* instrument driver with a C DLL in a Measurement Studio .NET application, use the Measurement Studio .NET Instrument Driver wizard to create .NET entry points to the C DLL functions you need to call from your application. Use the Add New Item wizard to select the .NET Instrument Driver Wizard.

The Measurement Studio .NET Instrument Driver wizard generates a .NET wrapper class for calling into IVI, VXI *plug&play*, and legacy instrument drivers based on the instrument driver function panel, header file, and an optional .sub file for IVI drivers. The wizard can generate both Visual C# and Visual Basic .NET source code. After completing the wizard, a new instrument driver wrapper class is added to your project and opened in the source code editor.



Tip For information about the .NET instrument driver wizard, refer to the *Calling Instrument Drivers from .NET Languages* topic in the *NI Measurement Studio Help*.

Measurement Studio .NET Class Libraries

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

Measurement Studio includes the following .NET class libraries:

- Analysis
- Common
- Network Variable
- Technical Data Management Streaming (TDMS)
- User Interface
 - Windows Forms
 - Web Forms
 - Windows Presentation Foundation (WPF)

The following .NET class libraries that interface with National Instruments device drivers are also available on the NI Device Drivers DVD or online from ni.com/drivers:

- NI-488.2
- NI-DAQmx
- NI-DCPower
- NI-Digital
- NI-DMM

- NI-FGEN
- NI-RFmx
- NI-RFPM
- NI-RFSA
- NI-RFSG
- NI-SCOPE
- NI-SWITCH
- NI-VISA

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

64-bit Support in Measurement Studio .NET Assemblies

You can use Measurement Studio .NET assemblies in 64-bit applications to take advantage of the increased processing power and memory capabilities that are available to 64-bit applications. For more information about working with 64-bit class libraries, review the following:

- Measurement Studio includes 64-bit support for all class libraries except for modular instruments .NET wrappers for other NI drivers. For a list of hardware class library support, refer to the *Driver Support* section.
- Not all NI DAQ hardware supported by NI-DAQmx is supported in 64-bit user mode. Refer to the *NI-DAQ Readme* for more information, installed at **Start»All Programs»National Instruments»NI-DAQ»NI-DAQ Readme**.

Analysis

The Measurement Studio Analysis .NET class library is in the `NationalInstruments.Analysis` namespace. The Analysis class library includes a set of classes that provides various digital signal processing, signal filtering, signal generation, peak detection, and other general mathematical functionality. Use this library to analyze acquired data or to generate data. Additionally, the documentation for the Analysis class library includes analysis code snippets that you can copy and paste into an application and use immediately.

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

Standard Analysis

The Standard Analysis class library, which ships with Measurement Studio Standard Edition, includes the sawtooth, sine, square, triangle, and basic function wave generators.

Professional Analysis

The Professional Analysis class library, which ships with Measurement Studio Professional Edition, includes the Standard Analysis functionality as well as the following functionality:

- Signal processing functions such as convolution, deconvolution, correlation, decimation, integration, and differentiation
- FFT, Inverse FFT, Real FFT, Fast Hartley, Inverse Fast Hartley, Fast Hilbert, Inverse Fast Hilbert, DST, Inverse DST, DCT, and Inverse DCT transformations
- Lowpass, Highpass, Bandpass, and Bandstop types of Bessel, Chebyshev, Inverse Chebyshev, Windowed, Kaiser, and Elliptic filters for real and complex data
- Narrowband FIR filters for real and complex data
- Linear algebra functions such as determinant, check positive definiteness, calculate dot product, and other various matrix functions
- Scaled and unscaled windowing classes for real and complex data
- Common statistical functions such as mean, median, mode, and variance
- Exponential, linear, and polynomial curve fitting functions
- Signal generation functions
- Basic level trigger detection

Enterprise Analysis

The Enterprise Analysis class library, which ships with Measurement Studio Enterprise Edition, includes the Standard and Professional Analysis functionality as well as the following advanced functionality:

- EquiRipple filters for real and complex data
- Linear algebra functions such as forward and back substitution, LU factorization, Cholesky factorization, Schur decomposition, and Hessenberg decomposition
- Probability and analysis of variance
- Sine, impulse, pulse, ramp, and chirp patterns
- General least square curve fit, power fit, log fit, Gauss fit, cubic spline fit, and interpolation functions
- Measurement functions such as transition measurements, pulse measurements, cycle RMS average, and single tone and multiple tone information
- Special functions

Refer to [Select Your Measurement Studio Edition](#) to determine the type of measurements available in the Professional and Enterprise Analysis .NET libraries.



Tip For more information about analyzing or generating data with the Analysis class library, refer to the *Using the Measurement Studio Analysis .NET Library* topic in the *NI Measurement Studio Help*.

Common

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

- A `ComplexSingle` data type. This data type represents a complex number of type `Single` that is composed of two single-precision floating-point numbers.
- A `ComplexDouble` data type. This data type represents a complex number of type `Double` that is composed of a real part and an imaginary part.
- A `ComplexInt16` data type. This data type represents a complex number of type `Int16` that is composed of a real part and an imaginary part.
- A `DigitalWaveform` data type. This data type represents a set of digital states that are grouped by samples or signals.
- A `ComplexWaveform` data type. This data type represents an analog signal that varies over time and is composed of complex data values.
- A `ComplexWaveformCollection` data type. This data type represents a strongly typed collection of `ComplexWaveform<TData>` objects. This data type encapsulates raw complex data, scaling information, and timing information.
- An `AnalogWaveform` data type. This data type represents an analog signal that varies over time.
- A `DataConverter` class that converts data from one data type to another data type, such as converting an array of integers to an array of doubles.
- An `EngineeringFormatInfo` class that defines a custom formatter to format numeric values as strings with engineering notation and International System of Units (SI) prefixes and symbols.
- A `PrecisionDateTime` structure that you can use to represent an instance in time, typically expressed as a date and time of day, that is accurate to the nearest 2^{-64} second.
- A `PrecisionTimeSpan` structure that you can use to represent a time interval, or duration of time, that is accurate to the nearest 2^{-64} second.
- A `PrecisionWaveformTiming` class that you can use to represent the timing of an analog or digital waveform that is accurate to the nearest 2^{-64} second.
- An `AnalogWaveformCollection` class that contains a strongly typed collection of `AnalogWaveform<TData>` objects; one object for each channel and record combination. You can access these objects through the 1D indexer or the 2D indexer.



Tip For more detailed information about the Common class library, refer to the *NI Measurement Studio Help*.

Data Transfer

You can use the `NetworkVariable` class library to transfer live measurement data between applications over a network. You can use `NetworkVariable` to exchange different types of data between Measurement Studio, LabVIEW, LabWindows/CVI, and other applications that support NI-Publish Subscribe Protocol (psp:). You can also use `NetworkVariable` to exchange

different types of data between OLE for Process Control (opc:) servers. Exchanging data between Measurement Studio applications and OPC servers with NetworkVariable requires the LabVIEW DSC Run-Time System.

Network Variable

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



Note Measurement Studio and LabWindows/CVI refer to variables as network variables and LabVIEW refers to variables as shared variables. However, you can read to and write from Measurement Studio and LabWindows/CVI network variables with LabVIEW shared variables.

- 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 `NationalInstruments.NetworkVariable.ServerProcess`, `NationalInstruments.NetworkVariable.ServerProcessInfo`, `NationalInstruments.NetworkVariable.ServerVariable`, and `NationalInstruments.NetworkVariable.ServerVariableInfo` classes to explicitly create network variables. You can configure the description and type for explicitly created network variables.
- 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 `WindowsForms` class.



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

Technical Data Management Streaming (TDMS)

Technical Data Management Streaming (TDMS) is a file format based on the National Instruments TDM data model used to stream data to disk. You can use the TDMS .NET class library to describe, store, and read measurement data that is optimized for high-speed data streaming and post processing. Additionally, you can use the TDMS .NET class library to

create files that you can use in LabVIEW, LabWindows/CVI, and DIAdem, and files created by these applications can be used by the TDMS .NET class library.

You can use the Measurement Studio TDMS .NET class library to perform the following operations:

- Read and write array data in a structured format from and to a .tdms file.
- Read and write analog waveform data or digital waveform data, including timing information, from and to a .tdms file.
- Use the `TdmsProperty` class to create custom properties for each level of the hierarchy by defining a name, data type, and value.



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

TDM Excel Add-In

You can use the TDM Excel Add-In to load NI .tdm and .tdms files into Microsoft Excel. Use the toolbar from within Excel to choose which properties are loaded into Excel at the file, group, and channel levels, including custom properties.

User Interface

The Measurement Studio user interface controls are in the Windows Presentation Foundation (WPF), Windows Forms, and Web Forms .NET class libraries. The following sections list the functionality included with the Measurement Studio WPF, Windows Forms, and Web Forms controls.

Windows Presentation Foundation (WPF) Controls

The WPF .NET class libraries are in the following namespaces:

- `NationalInstruments.Controls`
- `NationalInstruments.Controls.Converters`
- `NationalInstruments.Controls.Data`
- `NationalInstruments.Controls.Primitives`
- `NationalInstruments.Controls.Rendering`
- `NationalInstruments.DataInfrastructure`
- `NationalInstruments.DataInfrastructure.Descriptors`
- `NationalInstruments.DataInfrastructure.Primitives`
- `NationalInstruments.Logging`

The WPF class libraries encapsulate the following Measurement Studio user interface controls:

- **Graph**—Use the graph control to display 2D Cartesian data that requires two axes.
- **Writable graph**—Use the writable graph control to provide your end users with the ability to draw data into the graph.
- **Intensity graph**—Use the intensity graph control to display 3D data on a 2D surface, with color representing the third dimension.
- **Polar graph**—Use the polar graph to display 2D complex or point data.

- Digital Graph—Use this graph to plot and chart digital waveform data.
- Mixed Graph—Use this graph to display both analog and digital data on the same graph.
- Legend—Use the Legend control to display an icon and information associated with items in a graph, such as plots or cursors.
- Graph interaction palette—Use the GraphInteractionPalette to interact with a graph at run time
- Knob—Use Knob to set or indicate data values on a radial scale.
- Gauge—Use Gauge to set or indicate data values on a radial scale.
- Meter—Use Meter to set or indicate data values on a radial scale.
- Slider—Use Slider to set or indicate discrete data values on a vertical or horizontal scale.
- Tank—Use Tank to set or indicate cumulative data.
- Numeric Text Box—Use NumericTextBox to set or indicate one data value.
- Radix Numeric Text Box—Use RadixNumericTextbox to set or indicate data values in different numeric bases, such as hexadecimal or octal.
- Boolean Button—Use BooleanButton to display or interact with true and false data.
- Boolean Content Button—Use BooleanContentButton to provide separate content, such as different images and different text, for the true and false states.
- Arrow Button—Use ArrowButton to display or interact with true and false data using a customizable arrow-shaped button.
- Power Button—Use PowerButton to display or interact with true and false data using a button styled as a traditional computer power button.
- Switch—Use Switch to interact with Boolean input.
- LED—Use LED to indicate a Boolean value.
- TimeStamp TextBox—Use TimeStampTextBox to display and manipulate DateTime and PrecisionDateTime values as text and also enable use of a calendar to select dates.

Use these class libraries to add measurement-specific user interface controls to your application. You can configure the controls programmatically at design time with code or XAML or through the Properties window in the WPF design surface. The following sections describe each of the Measurement Studio WPF user interface controls.



Tip For more information about using the .NET user interface controls, refer to the *Using the Measurement Studio WPF .NET Controls* section in the *NI Measurement Studio Help*.

Windows Forms Controls

The Windows Forms .NET class library is in the `NationalInstruments.UI.WindowsForms` namespace. The Windows Forms class library encapsulates the following Measurement Studio user interface controls:

- Waveform graph—Use a waveform graph to plot waveform data.
- Scatter graph—Use a scatter graph to plot both x and y values.
- Digital waveform graph—Use a digital waveform graph to view the contents of a digital waveform, which encapsulates sampled digital signal data.
- Complex graph— Use a complex graph to plots `ComplexDouble` data.

- Intensity graph— Use the intensity graph to plot and chart three-dimensional data on a two-dimensional plot using color to represent the third dimension.
- Legend—Use the Legend control to display an icon and information associated with items in a graph, such as plots or cursors.
- Knob—Use Knob to set or indicate data values on a radial scale.
- Gauge—Use Gauge to set or indicate data values on a radial scale.
- Meter—Use Meter to set or indicate data values on a radial scale.
- Slide—Use Slide to set or indicate discrete data values on a vertical or horizontal scale.
- Thermometer— Use the thermometer control to display numeric values.
- Tank—Use the tank control to set or indicate cumulative data.
- Numeric edit— Use the numeric edit control to display and edit numeric values.
- Switch—Use the switch control to interact with Boolean input.
- LED—Use the LED control to indicate a Boolean value.
- Property editor— Use the property editor to edit property values at run time.
- Array controls— Use array controls to encapsulate other controls as an array.

Use the Windows Forms .NET class library to add measurement-specific user interface controls to your application. You can configure the controls programmatically at design time, through the Properties window in the Windows Forms Designer, or at run time with the property editor control.



Tip For more information about using the .NET user interface controls, refer to the *Using the Measurement Studio Windows Forms .NET Controls* section in the *NI Measurement Studio Help*.

ASP.NET Web Forms Controls

The Measurement Studio ASP.NET user interface controls are in the Web Forms .NET class library. The Web Forms .NET class library is in the `NationalInstruments.UI.WebForms` namespace. The Web Forms class library encapsulates the following Measurement Studio user interface controls:

- Waveform graph—Use a waveform graph to plot waveform data.
- Scatter graph—Use a scatter graph to plot both x and y values.
- Digital waveform graph—Use a digital waveform graph to view the contents of a digital waveform, which encapsulates sampled digital signal data.
- Complex graph— Use a complex graph to plots `ComplexDouble` data.
- Legend—Use the Legend control to display an icon and information associated with items in a graph, such as plots or cursors.
- Knob—Use Knob to set or indicate data values on a radial scale.
- Gauge—Use Gauge to set or indicate data values on a radial scale.
- Meter—Use Meter to set or indicate data values on a radial scale.
- Slide—Use Slide to set or indicate discrete data values on a vertical or horizontal scale.
- Thermometer— Use the thermometer control to display numeric values.
- Tank—Use the tank control to set or indicate cumulative data.
- Numeric edit— Use the numeric edit control to display and edit numeric values.
- Switch—Use the switch control to interact with Boolean input.

- LED—Use the LED control to indicate a Boolean value.
- AutoRefresh— Use to update a group of Measurement Studio ASP.NET Web Forms controls on a timed interval.



Note Measurement Studio Web Forms controls require complex UI functionality such as callbacks, view state management, and event handling to function properly. ASP.NET Model-View-Controller (MVC) does not include this functionality; therefore, Measurement Studio Web Forms controls do not support MVC.

Use this class library to add measurement-specific user interface controls to your Web application. You can configure the controls programmatically at design time or through the Properties window in the Web Forms Designer.

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*, accessible from within the Measurement Studio menu in the Visual Studio environment.

Where to Go Next

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

- Getting Started information—Refer to the installed *NI Measurement Studio Help* topic to get started with Measurement Studio. For a list of Measurement Studio resources, refer to *Using the Measurement Studio Help* in the *NI Measurement Studio Help*.
- Measurement Studio website, ni.com/mstudio—Contains Measurement Studio news, support, downloads, white papers, and product tutorials.
- NI Technical Support, ni.com/support—Provides access to online example programs, tutorials, technical news, and Measurement Studio discussion forums.
- Review the information from the Microsoft website on using Visual Studio.

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*.

Information is subject to change without notice. Refer to the *NI Trademarks and Logo Guidelines* at ni.com/trademarks for information on NI trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering NI products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at ni.com/patents. You can find information about end-user license agreements (EULAs) and third-party legal notices in the `readme` file for your NI product. Refer to the *Export Compliance Information* at ni.com/legal/export-compliance for the NI global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data. NI MAKES NO EXPRESS OR IMPLIED WARRANTIES AS TO THE ACCURACY OF THE INFORMATION CONTAINED HEREIN AND SHALL NOT BE LIABLE FOR ANY ERRORS. U.S. Government Customers: The data contained in this manual was developed at private expense and is subject to the applicable limited rights and restricted data rights as set forth in FAR 52.227-14, DFAR 252.227-7014, and DFAR 252.227-7015.

© 2010—2019 National Instruments. All rights reserved.

373225E-01 April 3, 2019