RELEASE NOTES

NI TestStand™

Version 4.1

These release notes contain NI TestStand 4.1 system requirements, installation instructions, information about new features, and other changes since TestStand 4.0. TestStand no longer includes a *Quick Start Guide*. Refer to the *Getting Started* section of this document instead.

Contents

Getting Started

The best way to familiarize yourself with TestStand is to explore the *Guide to TestStand Documentation* topic in the *NI TestStand Help*, which contains links to all the TestStand documentation in electronic format. Select **Start»All Programs»National Instruments»TestStand *x.x*» Documentation»NI TestStand Help** or select **Help»NI TestStand Help** in the TestStand Sequence Editor to access the *NI TestStand Help*.

Recommended System Requirements

National Instruments recommends the following system requirements to run TestStand 4.1. Minimum system requirements follow in parentheses.

- Microsoft Windows 2000 Service Pack 4, Windows XP Service Pack 2, or Windows Vista (32- and 64-bit) Business, Enterprise, or Ultimate Service Pack 1
- Pentium 4/M or equivalent processor (Pentium III, Celeron 866 MHz, or equivalent minimum)
- 1 GB of memory (256 MB minimum)
- 1.8 GB of free hard disk space
  - 1 GB of free hard disk space for TestStand
  - Additional 260 MB (x86) or 610 MB (x64) of free hard disk space for Microsoft .NET Framework Version 2.0 if not already installed
  - Additional 580 MB of free hard disk space for device drivers from the National Instruments Device Drivers CD
- SVGA resolution or higher video adapter (800 × 600 minimum video resolution for small fonts or 1024 × 768 minimum video resolution for large fonts)
- Microsoft Internet Explorer version 6.0 Service Pack 1 or later (version 5.5 minimum)
- Microsoft-compatible mouse

Use the following National Instruments application development environments (ADEs) with TestStand 4.1:

- LabVIEW 7.1.1 or later. You must use LabVIEW 8.2.1 or later to create custom sequence editors.

  (Windows Vista) LabVIEW 8.5 or later (LabVIEW 8.2.1 minimum)
- LabWindows™/CVI™ 7.1.1 or later. You must use LabWindows/CVI 8.1 or later to create custom sequence editors.

  (Windows Vista) LabWindows/CVI 8.5 or later (LabWindows/CVI 8.1.1 minimum). You must use

LabWindows/CVI 8.5.1 or later to create user interfaces for 64-bit Microsoft Vista.

- Measurement Studio 8.0.1 (or later) for integration with Microsoft Visual Studio 2005. You can use Measurement Studio 7.1 (or later) Enterprise Edition for integration with Visual Studio .NET 2003 or later, but TestStand can only call DLLs and .NET assemblies you create with Visual Studio .NET 2003. You must use Visual Studio 2005 to create and edit code directly from TestStand and to initiate debugging directly from TestStand.

  TestStand supports Visual Studio Standard but does not support Visual Studio Express. The .NET and MFC examples use projects and solutions created in Visual Studio 2005. If you are using an earlier version of Visual Studio, create new projects and solutions from the source files TestStand provides.

Installation Instructions

Note If you saved LabVIEW VIs that call the TestStand API with a version of LabVIEW earlier than 5.1.1, you must compile and save the VIs in LabVIEW 7.1.1 or later before you install TestStand 4.1. If you do not compile and save the VIs in LabVIEW 7.1.1 or later, the VIs do not load correctly after you install TestStand 4.1.

Unless you specify another location during installation, the TestStand installation program copies files to `<Program Files>\National Instruments\TestStand 4.1`.

National Instruments recommends exiting all programs before running the TestStand installer. Applications that run in the background, such as virus scanning utilities, might cause the installer to take longer than average to complete.

Complete the following steps to install TestStand.

1. Log on as an administrator or as a user with administrator privileges.
2. Insert the TestStand 4.1 installation CD and follow the instructions that appear on the screen.

National Instruments recommends installing the complete TestStand program. If you perform a custom installation and do not install all the TestStand features, you can run the installation program again later to install additional features.

If you install a version of LabVIEW later than LabVIEW 8.5 and you want to use that version of LabVIEW with TestStand, complete the following steps to update `TestStand - Default Values 85.llb` to the version of LabVIEW you want to use so the LabVIEW Adapter can retrieve the default parameter values of the VIs.

1. Create a copy of `TestStand - Default Values 85.llb`.
2. Rename the copy to `TestStand - Default Values` *`xx`*`.llb`, where *`xx`* indicates the version of LabVIEW you want to use with TestStand.
3. Mass compile `TestStand - Default Values` *`xx`*`.llb` using the version of LabVIEW you want to use with TestStand.

If you install LabVIEW, LabWindows/CVI, or Visual Studio after you install TestStand, launch the sequence editor or a user interface once so TestStand configures these applications to locate TestStand API files.

Refer to the National Instruments Web site at `ni.com/info` and enter the info code `rddrau` to access the latest software drivers and updates.

Activating TestStand Licenses

After you install TestStand, you must use the NI Activation Wizard to activate the software.

Note When you activate TestStand 4.1, you need the product serial number and the name of the product for the serial number. You can find both of these items on the proof-of-ownership and registration card you received with the product.

Refer to the *Activating Your Software* topic in the *NI TestStand Help* for more information about how to activate TestStand.

National Instruments offers a variety of TestStand licenses for the different ways you can use TestStand in development and deployment applications. In general, you can select from the following types of licenses: the TestStand Development System License, the TestStand Custom Sequence Editor License, the TestStand Debug Deployment Environment License, and the TestStand Base Deployment Engine License.

In most cases, when you first install TestStand, you activate a TestStand Development System License. Use the other licenses to activate TestStand on computers to which you deploy TestStand custom sequence editors or operator interfaces you develop.

Use the following descriptions only as a reference for the TestStand licensing options. Refer to `ni.com/activate` for more information about activating TestStand licenses. Refer to `ni.com/teststand` to purchase a TestStand license. Contact a local National Instruments representative for more information or for questions about specific licensing needs.

Note This document does not replace the National Instruments Software License Agreement installed in the `<National Instruments>\Shared\MDF\EULAs\NIReleased` directory.

TestStand Evaluation Package (350863-00)

When you run TestStand in Evaluation Mode, the software expires after 30 days. The software runs as a fully functional Development System for the first 7 days of the evaluation period. After 7 days, TestStand runs with the following restrictions:

- Sequence execution time limit of 10 minutes
- Continuous application usage time limit of 1 hour

You can activate a TestStand license at any point during or after the 30-day evaluation period.

TestStand Development System License (777777-03)

Activate the TestStand Development System License to develop and edit test sequences and to save TestStand sequence files within the TestStand Sequence Editor, within a TestStand Custom Sequence Editor, or programmatically using the TestStand API. This license also grants the ability to develop custom sequence editors and operator interfaces.

TestStand Custom Sequence Editor License (777775-01)

Activate this license to develop and edit test sequences and TestStand sequence files within a TestStand Custom Sequence Editor or programmatically using the TestStand API.

TestStand Debug Deployment Environment License (900851-01)

The TestStand Debug Deployment Environment License offers the most flexibility for deploying TestStand and LabVIEW-based, LabWindows/CVI-based, and Measurement Studio-based systems.

Activate this license to install the development versions of TestStand, LabVIEW, LabWindows/CVI, Measurement Studio, and any corresponding add-on toolkits on a single test station so you can debug deployed test applications on the test station. This license grants the ability to make minor edits to fix bugs in deployed test applications but does not grant the ability to perform any development tasks using TestStand, LabVIEW, LabWindows/CVI, or Measurement Studio on the test station.

TestStand Base Deployment Engine License (777774-09)

The TestStand Base Deployment Engine License is the minimum license required for all deployed TestStand-based applications. Activate this license to deploy the TestStand Engine, a TestStand Operator Interface, and TestStand sequence files to the single test station to which the license applies. This license does not grant the ability to perform any development tasks using the TestStand Sequence Editor, a TestStand Custom Sequence Editor, or the TestStand API.

Recommended Database Client Software

Use the following recommended database client software with the database components included with TestStand:

- **Microsoft Access**—Use the Microsoft Jet 4.0 OLE DB Provider.
- **Microsoft SQL Server**—Use the Microsoft OLE DB Provider for SQL Server or the SQL Native Client provider.
- **Oracle**—Use the latest Oracle Provider for OLE DB and Oracle Client software. You can download the Oracle Provider from the Oracle Web site at `www.oracle.com`.

Note National Instruments does not recommend using the Microsoft OLE DB Provider for Oracle because it does not support all the OLE DB features TestStand requires.

- **MySQL**—Use the MySQL ODBC Driver 3.51 or later.
- **Sybase SQL Anywhere**—Use the Adaptive Server Anywhere ODBC Driver 9.0 or later.

Installing Additional Software Components

TestStand 4.1 installs the following additional software components:

- .NET Framework 2.0
- LabVIEW 7.1.1 Run-Time Engine
- LabVIEW 8.0.1 Run-Time Engine
- LabVIEW 8.2.1 Run-Time Engine
- LabVIEW 8.5.1 Run-Time Engine
- LabWindows/CVI 8.5.1 Run-Time Engine
- LabWindows/CVI SQL Toolkit DLL version 2.1, `cvidb32.dll`
- National Instruments Session Manager

The National Instruments Device Driver CD contains the following suggested components:

- Measurement & Automation Explorer (MAX)
- IVI Compliance Package (ICP)

To use IVI, download and install IVI Compliant-specific drivers from the Instrument Driver Network, located at `ni.com/devzone/idnet`.

Installing Multiple Versions of TestStand on the Same Computer

You can install TestStand 4.1 on a computer that contains a previous TestStand version, but you cannot install TestStand 4.1 over a previous TestStand version. You must first uninstall a previous TestStand version to install TestStand 4.1 in the same directory.

Uninstalling TestStand 2.0.1 and Earlier

Because the uninstallers for TestStand 2.0.1 and earlier remove the `<TestStand>\OperatorInterfaces\User` directory, you must complete the following steps to safely uninstall the previous TestStand version and preserve all configuration files and files located in the `User` subdirectories.

1. Move the `<TestStand>\OperatorInterfaces\User` directory to a location outside the `<TestStand>` directory.
2. Navigate to **Add/Remove Programs** on the Windows Control Panel and select **National Instruments Software** to run the TestStand uninstaller. The TestStand uninstaller might launch a dialog box that requests confirmation to remove all TestStand configuration files and pre-installed user components. Click **No**.
3. When the TestStand uninstaller completes, move the `OperatorInterfaces\User` directory back into the original `<TestStand>` directory.

   You can now install TestStand 4.1 into this directory.

Using the TestStand Version Selector

Although you can install TestStand 4.1 on a computer that contains a previous TestStand version, only one version of TestStand can be active at a time. If you must install TestStand 4.1 on the same computer as a previous TestStand version, use the TestStand Version Selector to specify the active version of TestStand. Select **Start»All Programs»National Instruments»TestStand *x.x*»TestStand Version Selector** to launch the TestStand Version Selector application, `TSVerSelect.exe`, located in the `C:\Program Files\National Instruments\Shared\TestStand Version Selector` directory.

(Windows Vista) Windows Vista launches a User Account Control elevation prompt for you to manually resolve when you launch the TestStand Version Selector application.

If you activate TestStand 4.1 and run a user interface from the previous TestStand version, the user interface uses the TestStand 4.1 engine, step types, and components. If you activate the previous TestStand version and run a TestStand 4.1 User Interface or the TestStand Sequence Editor, those applications do not function correctly.

Directory Relocation

To comply with Windows Vista restrictions on writing to the `Program Files` directory and to improve usability for Windows 2000/XP users who do not have permission to write to the `Program Files` directory, TestStand 4.1 installs some files in different locations from previous versions of TestStand. Refer to the *TestStand and Windows Vista* section for more information about using TestStand on Windows Vista.

Table 1 lists the previous directory locations in TestStand 4.0 and the new locations in TestStand 4.1. The TestStand documentation and Table 1 refer to these directories in the following ways:

- **<TestStand>**—Located by default at `C:\Program Files\National Instruments\TestStand` *`x.x`* on Windows 2000/XP and Windows Vista (32-bit) and at `C:\Program Files (x86)\National Instruments\TestStand` *`x.x`* on Windows Vista (64-bit).
- **<TestStand Public>**—Located by default at `C:\Documents and Settings\All Users\Documents\National Instruments\TestStand` *`x.x`* on Windows 2000/XP and at `C:\Users\Public\Documents\National Instruments\TestStand` *`x.x`* on Windows Vista.

- **<TestStand Application Data>**—Hidden by default and located at `C:\Documents and Settings\All Users\Application Data\National Instruments\TestStand x.x` on Windows 2000/XP and at `C:\ProgramData\National Instruments\TestStand x.x` on Windows Vista.

Table 1. Directories Relocated in TestStand 4.1

Location in TestStand 4.0	Location in TestStand 4.1
`<TestStand>\AdapterSupport\CVI`	`<TestStand Public>\AdapterSupport\CVI`
`<TestStand>\AdapterSupport\LabVIEW`	`<TestStand Public>\AdapterSupport\LabVIEW`
`<TestStand>\Cfg`	`<TestStand Application Data>\Cfg`
`<TestStand>\CodeTemplates\NI`	`<TestStand>\CodeTemplates`
`<TestStand>\CodeTemplates\User`	`<TestStand Public>\CodeTemplates`
`<TestStand>\Components\NI`	`<TestStand>\Components`
`<TestStand>\Components\NI\RuntimeServers`	`<TestStand Public>\Components\RuntimeServers`
`<TestStand>\Components\User`	`<TestStand Public>\Components`
`<TestStand>\Examples`	`<TestStand Public>\Examples`
`<TestStand>\Setup`	`<TestStand Public>\Setup`
`<TestStand>\Tutorial`	`<TestStand Public>\Tutorial`
`<TestStand>\UserInterfaces\NI`	`<TestStand>\UserInterfaces`
`<TestStand>\UserInterfaces\User`	`<TestStand Public>\UserInterfaces`

Use the `<TestStand Public>\CodeTemplates`, `<TestStand Public>\Components`, and `<TestStand Public>\UserInterfaces` directories in place of the corresponding `User` directories in TestStand 4.0 and earlier. Use the `<TestStand>\CodeTemplates`, `<TestStand>\Components`, and `<TestStand>\UserInterfaces` directories in place of the corresponding `NI` directories in TestStand 4.0 and earlier. You can use the `Engine.GetTestStandPath` method to find the directories programmatically.

To modify the installed code templates or components or to create new code templates or components, copy the files from the `<TestStand>` directory to the `<TestStand Public>` directory and make changes to the copies. To modify the installed user interfaces or to create new user interfaces, modify the files TestStand installs in the `<TestStand Public>\UserInterfaces` directory. When you modify installed files, rename the files after you modify them if you want to create a separate custom component. You do not have to rename the files after you modify them if you only want to modify the behavior of an existing component. If you do not rename the files and you use the files in a future version of TestStand, changes National Instruments makes to the component might not be compatible with the modified version of the component. Storing new and customized files in the `<TestStand Public>` directory ensures that new installations of the same version of TestStand do not overwrite the customizations and ensures that uninstalling TestStand does not remove the files you customize.

The `AdapterSupport\CVI` and `AdapterSupport\LabVIEW` directories were moved to the `<TestStand Public>` directory because TestStand requires write access to these directories for standard users.

Migrating to TestStand 4.1

Complete the following tasks if you are migrating from a previous TestStand version.

- If you saved files in a relocated directory, copy the files to the new location in TestStand 4.1.
- Update projects and files that reference files and directories that have moved in TestStand 4.1. You can use the `Engine.GetTestStandPath` method to retrieve file and directory paths programmatically.

  If you use relative paths to reference files in the `<TestStand>` directory, the paths might break.

  If LabWindows/CVI returns a warning that some TestStand API files were not found when you open a project, remove the files from the project and re-add them from the `<CVI>\Common` directory in LabWindows/CVI 8.5 and later and from the `<CVI>\instr\TestStand\API\CVI` directory in LabWindows/CVI 8.1.1 or earlier.

  If Visual Studio returns a warning that some TestStand API or adapter support files were not found when you open a project, add the `$(TestStand)`, `$(TestStandAppData)`, and `$(TestStandPublic)` environment variables as needed to the Additional Include Directories control for the project.
- Use the Deployment Utility to update deployable images or installers for deployable images to redirect file destinations to the TestStand 4.1 locations. Select **View Destination** in the Distributed Files control and select the new location in the Installation Destination control in the Installer Properties section on the Distributed Files tab.

Migrating User Components

You can copy the following directories and files from the previous TestStand installation to the appropriate TestStand 4.1 directory to migrate configuration settings and user components. Refer to the *Directory Relocation* section for information about the TestStand 4.1 directory structure.

- `<TestStand>\Cfg\DocGen.ini`
- `<TestStand>\Cfg\StationGlobals.ini`
- `<TestStand>\Cfg\TestExec.ini`
- `<TestStand>\Cfg\TestStandDatabaseOptions.ini`
- `<TestStand>\Cfg\TestStandModelModelOptions.ini`
- `<TestStand>\Cfg\TestStandModelReportOptions.ini`
- `<TestStand>\Cfg\TypePalettes\MyTypes.ini` (TestStand 3.0 and earlier)
- `<TestStand>\Cfg\Users.ini`
- `<TestStand>\CodeTemplates\User`
- `<TestStand>\Components\User`
- `<TestStand>\OperatorInterfaces\User`
- `<TestStand>\UserInterfaces\User`

If you use custom Tools menu items in the previous TestStand version, complete the following steps to export the items from that installation and import them into TestStand 4.1.

1. In the previous TestStand version, select **Tools»Customize** in the TestStand Sequence Editor to launch the Customize Tools Menu dialog box.
2. Click the **Export Items to File** button to launch the Export Tools Menu dialog box.
3. Select the menu items to export to a Tools menu file and click **OK**.
4. Create a `<TestStand Public>\Setup\ToolMenusToInstall` directory.
5. Place the Tools menu file you created in step 3 in the new directory.
6. Launch the TestStand 4.1 Sequence Editor. TestStand adds the new menu items to the Tools menu and deletes the Tools menu file.

Migrating Changes to TestStand Components

TestStand includes several components you can customize, such as process models, user interfaces, and certain step types. If you made changes to one of these components, place the custom component in the appropriate `<TestStand Public>` directory after you install TestStand 4.1 and incorporate the TestStand 4.1 functionality.

If you made substantial or complex changes to the component, use a file-comparison tool to determine the changes between the TestStand 4.1 version of the component and the original version of the component you modified and apply the TestStand 4.1 improvements to the custom version of the component.

If you made minor changes to the component, use a file-comparison tool to determine the changes you made to the component and reapply the improvements to a copy of the TestStand 4.1 version of the component.

You can use the following types of file-comparison tools:

- To compare sequence files, select **Edit»Diff Sequence File With** in the TestStand Sequence Editor to use the TestStand Differ.
- To compare text files, use a source code comparison tool, such as Microsoft Windiff. You can also use the Diff command in the Edit menu of the LabWindows/CVI Source window.
- To compare VI files, select **Tools»Compare** in LabVIEW to use the Compare VIs tool. This tool is available only in the LabVIEW Professional Development System.

Note Subsets of different versions of the same component are not necessarily interoperable without modifications. For example, you cannot replace a single sequence in the TestStand 4.1 process models with the corresponding sequence from older TestStand process models without making further modifications. If you customized the TestStand process models, you must ensure that TestStand can find all the subordinate components the process models use and that any of those components that are ActiveX servers are registered. The default process model sequence for TestStand uses separate sequences, DLLs, and ActiveX servers to support database logging and report generation features.

Behavior Changes

TestStand includes the following behavior changes between version 4.0 and version 4.1. Refer to the *NI TestStand Help* for more information about behavior changes between earlier versions of TestStand. Refer to the *TestStand and Windows Vista* section for more information about behavior changes in Windows Vista that might affect applications that you write or modify.

- Use the Allow Automatic Type Conflict Resolution option on the Preferences tab of the Station Options dialog box to restrict automatic type conflict resolution. With the new default setting for this option, TestStand shows the type conflict dialog box in more cases than in previous versions of TestStand. National Instruments made this change to prevent unintended type propagation to type palette and sequence files. With the default setting, you can place types in a type palette file to ensure that TestStand does not automatically use a newer version of the types without prompting you. Refer to the *NI TestStand Help* for more information about changing the criteria for automatic type conflict resolution, including selecting the TestStand 4.0.*x* and earlier behavior.
- The IVI-C step types were renamed to IVI step types. The IVI step types now support IVI-COM class-compliant instrument drivers if you install the IVI-COM Adapter component of the IVI Component Package included on the NI Driver CD.
- `Logging.StepResultProperty` is deprecated. Instead, use `Logging.PropertyResult` to reference the property of a step result the database logger is processing. In addition, the context includes the `Logging.PropertyResultDetails` container, which contains information about the property `Logging.PropertyResult` references.
- TestStand now stores the customized settings of the Step Type Menu Editor in `<TestStand Application Data>\Cfg\TestExec.ini`. Previous TestStand versions stored these settings in type palette files, but these files are no longer writable on Windows Vista. The Step Type Menu Editor no longer marks the type palettes as modified.
- The `Engine.GetOutputMessages` method is deprecated. Do not call this method in new applications. To prevent unbounded memory use, the collection the `Engine.GetOutputMessages` method returns now automatically clears if the application fails to clear the collection in response to the UIMsg_OutputMessages event. Instead of calling the `Engine.GetOutputMessages` method, the application must obtain the newest output messages in the event callback by calling the `OutputMessages.CopyMessagesToCollection` method

the `UIMessage.ActiveXData` property holds for the UIMsg_OutputMessages event.

- The `OutputMessage.Post` method now posts a reference to the OutputMessage. Formerly, this method posted a copy of the `OutputMessage` object to which the caller of the `OutputMessage.Post` method had no direct access.
- TestStand 4.1 removes support for Microsoft Visual Basic 6.0. TestStand no longer ships with source code for components created with Visual Basic 6.0, including examples.
- The ActiveX/COM Adapter no longer generates an error when a server method or property call returns a positive HRESULT value.

What's New in TestStand 4.1

This section describes the new features in TestStand 4.1 and other changes since TestStand 4.0.

Sequence Hierarchy Window

TestStand sequences can use the Sequence Call step type to call subsequences. The new Sequence Hierarchy window shows the relationship between sequences and subsequences. Use the Sequence Hierarchy window to inspect and navigate a complex group of sequences to more easily modify, debug, and maintain test sequences. The Sequence Hierarchy window does not provide a way to edit Sequence Call steps and does not show execution flow while running. Use the following methods to access the Sequence Hierarchy window:

- Select **View»Sequence File»Display File Hierarchy** to display the hierarchy of all the sequences in the current sequence file or select **View»Sequence File»Display Hierarchy Using»<Entry Point Name>** to display the hierarchy from that entry point using the current sequence file as the client sequence file.
- Right-click the list of steps on the Steps pane of the Sequence File window and select **Display File Hierarchy** from the context menu to display the hierarchy of all sequences in the sequence file or **Display Hierarchy Using»<Entry Point Name>** from the context menu to display the hierarchy from that entry point using the current sequence file as a client sequence file.
- Right-click one or more sequences in the list of sequences on the Sequences pane of the Sequence File window and select **Display Sequence Hierarchy** from the context menu to display the hierarchy of the selected sequence or sequences, select **Display File Hierarchy** from the context menu to display the hierarchy of all the sequences in the sequence file, or select **Display Hierarchy Using»<Entry Point**

Name> from the context menu to display the hierarchy from that entry point using the current sequence file as a client sequence file.

If you do not select any sequences, the Display File Hierarchy option is available, and the Display Sequence Hierarchy option is disabled. If you select one or more sequences, the Display File Hierarchy is disabled, and the Display Sequence Hierarchy option is available.

- Right-click a sequence file in the list of sequence files on the Windows pane or on the Workspace pane and select **Display File Hierarchy** from the context menu to display the hierarchy of all sequences in the sequence file or **Display Hierarchy Using»<Entry Point Name>** from the context menu to display the hierarchy from that entry point using the current sequence file as a client sequence file.

Use the buttons on the Sequence Hierarchy toolbar to configure the display of the Sequence Hierarchy window.

Refer to the *NI TestStand Help* for more information about the Sequence Hierarchy window and about using the buttons on the Sequence Hierarchy toolbar.

Customizing Results

TestStand 4.1 includes an additional results table for every step so you can add arbitrary pieces of data to the result list and include that data in a report or log that data to a database. In the additional results table, enter a list of expressions or specify code module parameters you want to log. When the step executes, the step evaluates each expression and adds the result and parameter values to the `Locals.ResultList` property. You can log any number of data items with varying types.

Use the Additional Results panel on the Properties tab of the Step Settings pane in the TestStand Sequence Editor or use the Additional Results dialog box in a TestStand User Interface to access the additional results table.

Use the Additional Results step type when you want to log data outside the context of one particular step in the sequence. Use the Additional Results edit tab in the TestStand Sequence Editor and the Additional Results dialog box in a TestStand User Interface to configure the Additional Results step.

Refer to the *NI TestStand Help* for more information about the Additional Results panel, edit tab, and dialog box. Refer to the `AdditionalResults.seq` file located in the `<TestStand Public>\Examples\AdditionalResults` directory for an example that demonstrates how to use the additional results feature. This example adds variables, properties, expression values, and module call parameters to the result list and report.

Database Options Dialog Box Updates

TestStand 4.1 updates the Database Options dialog box in the following ways to support logging additional results:

- The Types to Log control on the Statements tab of the Database Options dialog box includes intrinsic data types and arrays of intrinsic data types.
- The Statements tab of the Database Options dialog box includes a new Traversing Options button that launches the Traversing Options dialog box, in which you can specify if subsequent statements can process the current UUT, the step or property result, or the sub-results of the step or property result.

Default Database Schema Updates

TestStand 4.1 updates the default database schema in the following ways to support logging additional results:

- The default database schema no longer includes the STEP_PASSFAIL, STEP_CALLEXE, STEP_MSGPOPUP, STEP_PROPERTYLOADER, STEP_STRINGVALUE, and MEAS_IVI_SINGLEPOINT tables. The default database schema now logs this information to the PROP_RESULT table, which includes property result information and simple intrinsic property values. The table contains a foreign key that references the STEP_RESULT table.
- The default database schema includes the PROP_BINARY table to log binary data, such as array values. The table contains a foreign key that references the PROP_RESULT table.
- The default database schema includes the PROP_ANALOGWAVEFORM table to log analog waveform data, such as the LabVIEWAnalogWaveform, NI_IviWave, and NI_IviWavePair data types. This table replaces the MEAS_IVI_WAVE and MEAS_IVI_WAVEPAIR tables in previous versions of TestStand. The table contains a foreign key that references the PROP_RESULT table.
- The default database schema includes the PROP_DIGITALWAVEFORM table to log digital waveform data, such as the LabVIEWDigitalWaveform data type. This table contains a foreign key that references the PROP_RESULT table.
- The default database schema includes the PROP_NUMERICLIMIT table to log numeric limit data. This table replaces the MEAS_NUMERICLIMIT table in previous versions of TestStand. This table contains a foreign key that references the PROP_RESULT table.

- The STEP_RESULT table includes new STEP_ID and CAUSED_SEQFAIL fields. The STEP_ID field specifies the unique step ID, and the CAUSED_SEQFAIL field specifies if a step caused a sequence failure.

Click the **Reload NI Schemas** button on the Schemas tab of the Database Options dialog box to use previous versions of the default database schemas.

Resource Usage Profiler

Select **Tools»Profile Resource Usage** to launch the Resource Usage Profiler window to view and record the resources a multithreaded TestStand system uses over a period of time. The profiler records resource usage and TestStand thread synchronization operations the system performs as long as the Resource Usage Profiler window is open.

You can review the recorded data in graphs and sortable tables to identify performance bottlenecks and design flaws and to gain insight into the behavior and timing of complex multithreaded systems. You can copy the information to external applications, such as Microsoft Word or Excel.

Refer to the `Comparing Resource Usage Strategies.seq` file in the `<TestStand Public>\Examples\ResourceUsageProfiler` directory for an example of how to use the profiler. The example sequence file automatically launches the profiler and displays instructions for the example. Refer to the *NI TestStand Help* for more information about the Resource Usage Profiler.

TestStand and Windows Vista

TestStand 4.1 runs on Windows Vista.

Windows Vista Behavior Change

Windows Vista introduced a change in behavior that can cause applications to receive WM_PAINT messages unexpectedly while performing ActiveX/COM calls from .NET components or .NET applications. National Instruments has tested TestStand 4.1 and fixed instances where this behavior change caused the TestStand Sequence Editor and TestStand User Interface controls to behave incorrectly. Refer to the National Instruments KnowledgeBase at `ni.com/info` and enter the info code `rdwttc` for more information about this Windows Vista behavior change and how it might affect applications that you write or modify.

User Account Control Elevation Prompts

The User Account Control (UAC) security component in Windows Vista requires administrator privileges for some tasks, such as installing software, running certain applications, and changing system settings. If you are logged in as a standard user, Vista launches a UAC elevation prompt for prohibited tasks. You cannot resolve the UAC elevation prompts programmatically. Refer to Microsoft documentation for more information about UAC prompts.

TestStand 4.1 uses a Windows service to automatically handle most TestStand-related UAC prompts and notifications. The National Instruments TestStand Service runs with administrator privileges in the background and is responsible for tasks that require administrator privileges. The TestStand Service does not automatically handle applying settings for remote execution and applying settings and performing actions the TestStand Version Selector application requires. Performing these actions while logged on to Windows Vista as a user with standard privileges results in a UAC elevation prompt.

Authenticode Signatures

Authenticode signatures can help identify the publisher of a binary file and can help ensure that a binary file has not been modified since publication. Refer to Microsoft documentation for more information about Authenticode signatures.

Add an Authenticode signature to a TestStand user interface you create if you plan to allow users to download the user interface from a non-trusted public site and you want the operating system to identify your company as the publisher of the user interface. Also add an Authenticode signature to a user interface you create if the user interface requires administrator privileges to run on Windows Vista and you want the UAC elevation prompt to identify your company as the publisher of the user interface.

To verify an Authenticode signature, the requesting computer must connect to the Internet to obtain a current Certificate Revocation List (CRL). For .NET applications, the .NET Common Language Runtime (CLR) verifies Authenticode signatures for assemblies. If the computer that loads the assembly is not connected to the Internet, the CLR waits 15 seconds before timing out.

Complete the following steps to disable CRL validation in Internet Explorer to avoid the timeout period on the computer, even if the default browser on the computer is not Internet Explorer. Using the Internet Explorer Internet Options to disable CRL validation does not expose the computer to any additional security threats.

1. Navigate to **Internet Options** on the Windows Control Panel and click the Advanced tab.
2. In the Security section, disable the **Check for publisher's certificate revocation** option.

Alternatively, you can disable CRL validation by setting the registry key value of `HKCU\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing\State` to `0x00023e00`. To enable CRL validation, set the registry key value to `0x00023c00`.

When you disable CRL validation to avoid the timeout period, the CLR does not validate Authenticode-signed assemblies and does not grant the assemblies publisher evidence or publisher identity permissions, which is the same result when a timeout occurs. If the assemblies need these permissions, the computer must connect to the Internet or you must download a current CRL every 10–15 days.

As an alternative to disabling CRL validation for the entire computer, you can work around CRL validation if an application that uses the .NET Framework 2.0 and that has an Authenticode signature experiences the 15-second load time delay. Microsoft provides a fix you can download so you can correct this delay for .NET Framework 2.0 applications. The .NET Framework 2.0 Service Pack 1 also includes this fix. Refer to the Microsoft Knowledge Base article 936707 at `support.microsoft.com/kb/936707` for more information about correcting delays in .NET Framework 2.0 applications that use Authenticode signatures.

The TestStand Sequence Editor and user interface examples do not include Authenticode signatures because National Instruments distributes TestStand through trusted channels and because the TestStand Sequence Editor and user interface examples do not require administrator privileges to run on Windows Vista. Additionally, National Instruments finds the 15-second load time delay on isolated networks unacceptable and believes that you should use discretion when disabling CRL validation. Therefore, if you run the sequence editor or example user interfaces as administrator on Windows Vista, the UAC elevation prompt does not identify the sequence editor or example user interface as a National Instruments product.

Application Manifests

When an application launches on Windows Vista, the UAC security component determines whether to grant the application administrative privileges. A user that logs into Windows Vista as a standard user can write only to specific locations on disk and in the registry. Standard user is the default login for Windows Vista.

Microsoft recommends that applications run without requiring administrator privileges. If you design applications that do not attempt to access protected areas of the operating system, all users can run the application as intended without requiring administrator privileges. You can also include manifests to specify the execution level the application requires.

If an application does not specify an execution level in its manifest, the UAC launches the application with the standard or administrator privileges of the user. With standard privileges, the system uses virtualization to redirect any read and write operations for system files and registry keys to a per-user location instead of the actual system copy of the file or registry key. Do not create applications that rely on virtualization to perform these types of administrative operations.

The default TestStand user interface application binary files include manifests that instruct the UAC to execute the application without virtualization and without requiring administrative privileges. LabVIEW 8.5 and later automatically include a default manifest in built applications. LabWindows/CVI 8.5 and later allow you to specify a manifest for built applications. When you build the application, refer to the documentation for the ADE you used for more information about how to include a manifest.

Visual Studio Integration

When you launch Visual Studio 2005 from TestStand, Visual Studio runs with the same privileges you used to run TestStand. If you launch TestStand, log in as a user with standard privileges, and launch Visual Studio from TestStand, you cannot execute tasks in Visual Studio that require administrator privileges.

HTBasic Support

HTBasic currently does not support Windows Vista. If you installed the HTBasic 9.0 development environment on Windows Vista, you can still perform the Edit Subroutine and Create Subroutine functions on the HTBasic Module tab in the TestStand Sequence Editor when you use a step configured to use the HTBasic Adapter. However, HTBasic code modules might not run correctly.

API Additions

Refer to the *TestStand 4.1 API and UI Controls Additions and Changes* topic in the *NI TestStand Help* for more information about new API classes, methods, and properties. Additionally, some API methods and properties are now obsolete. The *NI TestStand Help* also documents API changes and additions for some earlier versions of TestStand.

Note Some components of TestStand, such as the TestStand Sequence Editor, expose .NET assemblies; however, the use of undocumented assembly API is not supported.

Other TestStand Enhancements

TestStand 4.1 includes the following other enhancements:

- The Synchronization step types and LabVIEW Utility step types now include step settings panels, which you use to configure all Synchronization and LabVIEW Utility step types in the TestStand Sequence Editor. Use the configuration dialog boxes to configure these step types in a TestStand User Interface.
- Use the new Tcl step types, located in the `<TestStand Public>\Examples\Tcl` directory, to call Tcl scripts from TestStand so you can execute Tcl code, get and set variables, and set and retrieve data in the console window for use in TestStand.
- Select **Tools»Update VI Calls** to run the Update VI Calls tool to update the Express VIs a LabVIEW step instance calls and to check or update a Standard VI call prototype. Use the Update VI Calls tool when you upgrade the LabVIEW version and you want to run the Express VIs a LabVIEW step instance calls in the new version of the LabVIEW Run-Time Engine. Also, use the Update VI Calls tool to update existing Express VI instances with any changes made to the Express VI. Refer to the *NI TestStand Help* for more information about the Update VI Calls tool.
- Use the Allow Automatic Type Conflict Resolution option on the Preferences tab of the Station Options dialog box to restrict automatic type conflict resolution. With the default setting for this option, TestStand shows the type conflict dialog box in more cases than in previous versions of TestStand. National Instruments made this change to prevent unintended type propagation to type palette and sequence files. With the default setting, you can place types in a type palette file to ensure that TestStand does not automatically use a newer version of the types without prompting you. Refer to the *NI TestStand Help* for more information about changing the criteria for automatic type conflict resolution.

- You can save XML or binary sequence files so they are compatible with TestStand 4.0. However, if you save a sequence file as any version earlier than TestStand 4.0, the output format is always INI because INI is the only sequence file format previous versions of TestStand supported.
- The *NI TestStand Help* and manuals include updated content. The help files are located in the `<TestStand>\Doc\Help` directory. The manuals are located in the `<TestStand>\Doc\Manuals` directory.

Note If you open help files directly from the `<TestStand>\Doc\Help` directory, National Instruments recommends that you open `TSHelp.chm` first because this file is a collection of all the TestStand help files and provides a complete table of contents and index.

372519M-01 May08