RELEASE NOTES

NI TestStand™

Version 4.2

These release notes contain NI TestStand 4.2 system requirements, installation instructions, information about new features, and other changes since TestStand 4.1.

Contents

Getting Started 2
- Recommended System Requirements 2
- Installation Instructions 3
- Activating TestStand Licenses 4
- Recommended Database Client Software 7
- Installing Additional Software Components 7
- Installing Multiple Versions of TestStand on the Same Computer 8
- TestStand and Windows Vista 8
- Using the TestStand Version Selector 10
- Directory Relocation 10
- Migrating to TestStand 4.2 12
- Migrating User Components 13
- Migrating Changes to TestStand Components 13

Behavior Changes 14

What's New in TestStand 4.2 15
- TestStand Deployment Utility Improvements 15
- Using Expressions to Customize Report File Paths 16
- On-The-Fly XML Report Improvements 18
- Improved XML Style Sheets 18
- Call Executable Step Improvements for Scripting Languages 18
- Visual Studio 2008 Support 19
- Customizing TestStand Behavior on Multi-core Systems 20
- PropertyObject Attribute Support 21
- Setting the Policy for Marking Files as Modified 22
- API Additions 23
- Other TestStand Enhancements 23

NATIONAL INSTRUMENTS™

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*.

Note TestStand no longer includes the *Using LabVIEW with TestStand* and *Using LabWindows/CVI with TestStand* manuals. Refer to the new *Using LabVIEW and LabWindows/CVI with TestStand* manual for the same content.

Note The *NI TestStand Reference Manual* is available only as a PDF file, located at `<TestStand>\Doc\Manuals\TestStandReferenceManual.pdf`.

Recommended System Requirements

National Instruments recommends the following system requirements to run TestStand 4.2. 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 GB minimum 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 free hard disk space for device drivers from the National Instruments Device Drivers DVD. The amount of free hard disk space required varies according to the drivers you choose to install. Install only the drivers for the hardware you will use.
- 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 4.2:

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

  (Windows Vista) LabVIEW 8.5.1 or later (LabVIEW 8.2.1 minimum)

- LabWindows™/CVI™ 8.0.1 or later. You must use LabWindows/CVI 8.1.1 or later to create custom sequence editors.

  (Windows Vista) LabWindows/CVI 8.5.1 or later (LabWindows/CVI 8.1.1 minimum). You must use LabWindows/CVI 8.5.1 or later to create user interfaces for Microsoft Vista (64-bit).

- Measurement Studio 8.0.1 (or later) for integration with Microsoft Visual Studio 2005 and Measurement Studio 8.5 (or later) for integration with Visual Studio 2008.

  TestStand supports integration with Visual Studio Standard but does not support integration with 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.

Note TestStand can execute code modules developed with versions of ADEs earlier than the listed supported versions, but National Instruments performs only limited testing with earlier versions of ADEs.

Installation Instructions

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

Note You cannot install TestStand to a network path or a mapped network path. You must install TestStand on a local computer.

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 4.2 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.

When you install a version of LabVIEW later than LabVIEW 8.6.*x* and you want to use that version of LabVIEW with TestStand, you must complete the following steps to update `TestStand - Default Values 86.llb` to allow the LabVIEW Adapter to retrieve the default parameter values of the VIs.

1. Create a copy of `TestStand - Default Values 86.llb`.
2. Rename the copy to `TestStand - Default Values xx.llb`, where *xx* indicates the version of LabVIEW, without a revision number, 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 or initiate the evaluation period for the software. When you activate TestStand, you need the serial number and the name of the software kit. You can find both of these items on the Certificate of Ownership card included in your software kit. 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. 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.

Refer to the *Licensing Options for TestStand Systems* section of Chapter 1, *Introduction to TestStand*, of the *Using TestStand* manual for information about how to use the available licenses when you build a TestStand-based test solution.

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

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 (777777-09)

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 (779851-09)

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.

You cannot activate and deactivate the TestStand Debug Deployment Environment License and reuse it on multiple computers. If you need to use a single debug license across multiple computers, contact National Instruments for more information about the Concurrent TestStand Debug Deployment Environment License.

TestStand Base Deployment Engine License (777774-03)

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 Object Linking and Embedding Database (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 Open Database Connectivity (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 includes the device driver software on a DVD. If you require device driver software on CDs, refer to the National Instruments Web site at `ni.com/info` and enter the info code `drivercd`.

TestStand 4.2 installs the following additional software components:

- .NET Framework 2.0
- LabVIEW 7.1.1 Run-Time Engine
- LabVIEW 8.2.1 Run-Time Engine
- LabVIEW 8.5.1 Run-Time Engine
- LabVIEW 8.6.1 Run-Time Engine
- LabWindows/CVI 9.0.1 Run-Time Engine
- LabWindows/CVI SQL Toolkit DLL version 2.1, `cvidb32.dll`
- National Instruments Session Manager

Note TestStand can use newer versions of the LabVIEW Run-Time Engine when you install LabVIEW on your development system. You can include newer versions of the LabVIEW Run-Time Engine in deployments using the Drivers and Components dialog box of the TestStand Deployment Utility. Refer to the *NI TestStand Help* for more information about the Drivers and Components dialog box.

The National Instruments Device Driver DVD 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 4.2 on a computer that contains a previous TestStand version, but you cannot install TestStand 4.2 over a previous TestStand version. You must first uninstall a previous TestStand version to install TestStand 4.2 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. If the TestStand uninstaller launches 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 4.2 into this directory.

TestStand and Windows Vista

TestStand 4.2 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 4.2 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 user with standard privileges, 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 4.2 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. Performing these actions while logged in to Windows Vista as a user with standard privileges results in a UAC elevation prompt.

Visual Studio Integration

When you launch Visual Studio from TestStand, Visual Studio runs with the same privileges you used to run TestStand. If you launch TestStand while logged in as a user with standard privileges and you launch Visual Studio from TestStand, you cannot execute tasks in Visual Studio that require administrator privileges. Refer to the *Visual Studio 2008 Support* section of this document for more information about using Visual Studio 2008 with TestStand.

Using the TestStand Version Selector

Although you can install TestStand 4.2 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 4.2 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.

If you activate TestStand 4.2 and run a TestStand User Interface from the previous TestStand version, the user interface uses the TestStand 4.2 engine, step types, and components. If you activate the previous TestStand version and run a TestStand 4.2 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 or later 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 or later. 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 or later
<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

Note 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.

Use the <TestStand Public>\CodeTemplates, <TestStand Public>\Components, and <TestStand Public>\UserInterfaces directories in place of the corresponding User directories in TestStand 4.0 or earlier. Use the <TestStand>\CodeTemplates, <TestStand>\Components, and <TestStand>\UserInterfaces directories in place of the corresponding NI directories in TestStand 4.0 or 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.

Migrating to TestStand 4.2

If you are migrating from a version of TestStand earlier than TestStand 4.1, complete the following tasks:

- If you saved files in a relocated directory, copy the files to the new location in TestStand 4.1 or later.
- Update projects and files that reference files and directories that have moved in TestStand 4.1 or later. 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 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 or 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 TestStand Deployment Utility to update deployable images or installers for deployable images to redirect file destinations to the TestStand 4.1 or later 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 4.2 directory to migrate configuration settings and user components. If you are migrating from a version of TestStand earlier than TestStand 4.1, refer to the *Directory Relocation* section for more information about directory location changes.

- `<TestStand Application Data>\Cfg\StationGlobals.ini`
- `<TestStand Application Data>\Cfg\TestExec.ini`
- `<TestStand Application Data>\Cfg\TestStandModelReportOptions.ini`
- `<TestStand Application Data>\Cfg\Users.ini`
- `<TestStand Public>\CodeTemplates`
- `<TestStand Public>\Components`
- `<TestStand Public>\UserInterfaces`

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 4.2.

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 4.2 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 4.2.

If you made substantial or complex changes to the component, use a file-comparison tool to determine the changes between the TestStand 4.2 version of the component and the original version of the component you modified, and to apply the TestStand 4.2 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 to reapply the improvements to a copy of the TestStand 4.2 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 or select **Start»All Programs»National Instruments»TestStand *x.x*»Tools»Sequence File Differ** 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 4.2 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 4.1 and version 4.2. 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.

- The constant `TSConst_TestStandPath_PublicComponent` in the `TSEnum_TestStandPaths` enumeration in `tsapicvi.h`, located in the `<TestStand>\API\CVI` directory, has been renamed `TS_TestStandPath_PublicComponent` to match the convention used in the file. This change will break any code that used the incorrectly named constant, but only when the code is recompiled.

What's New in TestStand 4.2

This section describes the new features in TestStand 4.2 and other changes since TestStand 4.1. This section only summarizes each new feature. Refer to the TestStand documentation listed for a particular feature for more information about the feature.

TestStand Deployment Utility Improvements

The TestStand Deployment Utility now uses improved internal caching to accelerate analysis and build times. Additionally, the LabVIEW VI Options dialog box includes the following new options so you can choose between faster performance and improved error handling:

- **Check for Broken VIs During Analysis**—When enabled, the TestStand Deployment Utility checks for broken VIs during analysis. This option is disabled by default to improve performance. Enable this option to debug VI errors.
- **Check for Broken VIs After Build**—When enabled, the TestStand Deployment Utility checks for broken VIs in the image directory when building an image. This option is enabled by default to ensure that distributed files do not contain broken VIs. Disable this option for better performance if you are rebuilding and have not made changes to VIs.
- **Remove Unused SubVIs**—Removes unused polymorphic subVIs and disconnects type definitions when building. This option is disabled by default for backward compatibility. Enable this option for better performance and to deliver a smaller number of subVIs.

Note Enabling the Remove Unused SubVIs option disables certain editing and debugging options on the deployed system, such as wiring a new type to a polymorphic VI or making type changes across several VIs by editing a type definition.

The Deployment Utility also better integrates with LabVIEW 8.6.1 or later for improved build speeds and deployment of advanced LabVIEW features, such as VIs that include LabVIEW object-oriented programming.

The Installer Properties section of the Distributed Files tab of the TestStand Deployment Utility also includes a new Force File to Install option. Enable this option to force the installers the deployment utility builds to always overwrite existing files.

Refer to Chapter 14, *Deploying TestStand Systems*, of the *NI TestStand Reference Manual* and to the *NI TestStand Help* for more information about the TestStand Deployment Utility.

Using Expressions to Customize Report File Paths

In addition to the existing options to customize reports, you can now use predefined macros and custom expressions on the Report File Pathname tab of the Report Options dialog box to specify the filename of reports and the directory in which TestStand stores reports. For example, you can use expressions to save reports in a directory that uses the sequence filename, save reports in different directories based on execution status, and generate report filenames that include the serial number, user name, or execution status. You can use expressions to replicate all existing report filename options.

Select **Specify Report File Path by Expression** in the File Name/Directory Options drop-down list in the Generate Report File Path section on the Report File Pathname tab of the Report Options dialog box to enter an expression in the Report File Path control. Table 2 lists common report options and the equivalent expression for the Sequential model using the client sequence file directory as the base directory.

Table 2. Current Report Options and Equivalent Expressions

Current Report Option	Equivalent Expression
New UUT Report File for Each UUT	`<ClientFileDir>\Report_<UUT>.<FileExtension>`
Prefix Sequence File Name to Report File Name	`<ClientFileDir>\<ClientFileName>_Report.<FileExtension>`
Add Date and Time to File Name	`<ClientFileDir>\Report_<FileDate><FileTime>.<FileExtension>`
Force File Name to Be Unique	`<ClientFileDir>\Report_<Unique>.<FileExtension>`
Prefix Sequence File Name to Report File Name + New UUT Report File for Each UUT + Add Date and Time to File Name	`<ClientFileDir>\<ClientFileName>_Report_<UUT>_<FileDate><FileTime>.<FileExtension>`

Table 3 lists example expressions for common tasks using the Sequential model.

Table 3. Common Tasks and Equivalent Expressions

Task	Equivalent Expression
Save reports in a directory that uses the sequence filename	`C:\<ClientFileName>\Report.<FileExtension>` **Note**: TestStand overwrites existing files unless you use the `<Unique>` macro or enable the Append if File Already Exists option.
Save reports in different directories based on execution status	`C:\<ClientFileName>\<UUTStatus>\Report.<FileExtension>` **Note**: TestStand overwrites existing files unless you use the `<Unique>` macro or enable the Append if File Already Exists option.

Table 3. Common Tasks and Equivalent Expressions (Continued)

Task	Equivalent Expression
Generate report filenames that include the UUT serial number	C:\<ClientFileName>\Report_<UUT>.<FileExtension>
Generate report filenames that include the user name	C:\<ClientFileName>\Report_<UserName>.<FileExtension>
Generate report filenames that include the execution status	C:\<ClientFileName>\Report_<UUTStatus>.<FileExtension>

TestStand supports the following macros:

- <Batch>—Batch serial number the user provides during execution. If the batch serial number is empty, TestStand uses the Empty_Batch_SerialNo resource string.
- <ClientFileDir>—Directory that contains the client sequence file.
- <ClientFileName>—Name of the client sequence file.
- <Desktop>—Directory path of the desktop.
- <FileExtension>—File format extension that corresponds to the report format you specify on the Contents tab of the Report Options dialog box.
- <FileDate>—String that contains the date when TestStand first writes the file to disk during execution.
- <FileDay>—String that contains the numeric day (1 through 31) when TestStand first writes the file to disk during execution.
- <FileMonth>—String that contains the numeric month (1 through 12) when TestStand first writes the file to disk during execution.
- <FileTime>—String that contains the time when TestStand first writes the file to disk during execution.
- <FileYear>—String that contains the year when TestStand first writes the file to disk during execution.
- <PublicDocumentsDir>—Public documents directory.
- <StationID>—Station ID during execution.
- <TempDir>—Temporary directory.
- <TestSocket>—Test socket number during execution. This value is always 0 for the Sequential model.
- <TestStandExecutableDir>—Directory of the TestStand executable.
- <TestStandPublicDir>—<TestStand Public> directory.
- <Unique>—A unique numerical value if the path evaluated already exists on disk.

- `<UserDocumentsDir>`—`My Documents` directory.
- `<UserName>`—Username currently logged in during execution.
- `<UUT>`—UUT serial number users specify during execution. If the serial number is empty, TestStand uses the `Empty_UUT_SerialNo` resource string.
- `<UUTStatus>`—UUT status during execution.

Refer to the *Specifying Report File Paths by Expression* topic in the *NI TestStand Help* for more information about each macro and about using expressions to specify the filename of reports.

On-The-Fly XML Report Improvements

TestStand includes improved style sheets and uses new report API objects to generate on-the-fly XML reports more efficiently. Refer to the *XML Reports* section of Chapter 6, *Database Logging and Report Generation*, of the *NI TestStand Reference Manual* for more information about XML reports.

Improved XML Style Sheets

TestStand includes restructured style sheets that reduce the time required to generate HTML output from XML report files and to simplify the task of modifying style sheets so that you can change the appearance and content of XML reports. Refer to the *XML Report Style Sheets* topic in the *NI TestStand Help* for more information about customizing TestStand XML report style sheets and using the included XML schemas. Refer to the *XML Reports* section of Chapter 6, *Database Logging and Report Generation*, of the *NI TestStand Reference Manual* for more information about XML reports.

Call Executable Step Improvements for Scripting Languages

You can now use the Call Executable step to more easily call scripting languages, such as Python or Perl, using any of the following methods:

- Specifying the script file as the file to execute
- Configuring the standard input text or file to pass to the executable call
- Configuring the file or variable to retrieve standard output and error
- Specifying the working directory

You can now use the File Pathname control on the Call Settings tab of the Call Executable Step Settings pane to specify the name of a file to open using the default application associated with the extension of the file. For example, you can launch a `.bat` file or open a `.txt` file by specifying the non-executable file directly. You can also use expressions to specify the path to the file.

You must configure the step to wait for the specified executable call to exit or wait for a specified time to enable access to the options you use to specify standard input to pass to the call and to specify how to retrieve standard output and standard error from the call.

You can configure the step to stream a file as input or evaluate a string or an expression as input. You can store the output and error information in a file on the local computer or on the remote computer when you call the executable on a remote computer. TestStand overwrites the file if it already exists. Alternatively, you can use an expression to store the output and error information in a TestStand variable or property, and you can use the status expression of the step to evaluate the output value. You can configure the step to ignore or set the step status to Error or Failed when the executable you call returns an error. You can include error information in the report TestStand generates.

You can specify the working directory for the executable by configuring TestStand to use the file directory, the sequence file directory, the current directory for the process, or an absolute path. You can also use expressions to determine the working directory at run time.

Refer to Chapter 4, *Built-In Step Types*, of the *NI TestStand Reference Manual* and to the *NI TestStand Help* for more information about the Call Executable step type.

Visual Studio 2008 Support

TestStand now supports creating, editing, and debugging code modules with Visual Studio 2008, in addition to existing support for Visual Studio 2005. TestStand no longer supports using Visual Studio 2003 for these operations from within TestStand. You can call DLLs or manually attach to a TestStand process to debug native code using Visual Studio 2003 or earlier. However, you cannot attach to a TestStand process using Visual Studio 2003 if you want to debug managed code.

Use the .NET Adapter Configuration and the C/C++ DLL Adapter Configuration dialog boxes to specify the versions of Visual Studio you want to use when you create or edit code, or when you debug code. You can configure the .NET and C/C++ DLL Adapters to use a specific version of Visual Studio installed on the computer, to use the version that matches the project file the step specifies, or to prompt you when performing the operation.

Refer to Chapter 5, *Module Adapters*, of the *NI TestStand Reference Manual* for more information about using Visual Studio with TestStand.

Customizing TestStand Behavior on Multi-core Systems

Symmetric multiprocessing (SMP) systems are computers that contain multiple CPUs, where a CPU can be a single core on a multi-core processor or a single-core processor on a multiprocessor computer. Windows and other modern operating systems can take advantage of SMP systems to achieve increased performance. SMP systems achieve better performance by executing multiple threads on multiple CPUs concurrently. Performance improvement can come through executing multiple processes concurrently as well as from executing multiple threads within a process concurrently. Generally, when an application is implemented with multiple threads, the operating system attempts to schedule each thread on a separate CPU when possible. Under some circumstances, this automatic scheduling can have a negative impact on application performance.

For instances in which the automatic scheduling of threads on CPUs negatively affects TestStand application performance, you can manually assign or constrain the TestStand process or specific threads to specific CPUs using the following new TestStand features:

- **Station Option: Default CPU Affinity for Threads**—Determines the default CPU affinity TestStand gives new threads of executions and the user interface thread. The default behavior is to allow all CPUs. This option is located on the Preferences tab of the Station Options dialog box.
- **Sequence Call Step Setting: CPU Affinity for New Thread or Execution**—Specifies the CPU affinity of the new thread in the current execution or the initial thread in a new execution. By default, the new thread uses the Default CPU Affinity for Threads station option. However, the new thread can also allow all CPUs, use the CPU affinity of the calling sequence, or allow specific CPUs that you specify. This option is located on the Sequence Call Advanced Settings window.
- **CPU Affinity Step Type**—Directly gets or sets the CPU affinity of the process or the current execution thread. This step provides low-level control over how a system utilizes CPUs, and is located in the Advanced subgroup of the Synchronization group on the Insertion Palette or in the Insert Step submenu of the Steps pane context menu. Refer to Appendix B, *Synchronization Step Types*, of the *NI TestStand Reference Manual* and to the *NI TestStand Help* for more information about the CPU Affinity step type.

Refer to the *Using TestStand on SMP Systems* topic in the *NI TestStand Help* for more information about optimizing TestStand performance on SMP systems for multithreaded applications and about when you might need to use these new features to control CPU affinity in TestStand.

PropertyObject Attribute Support

You can now create attributes for TestStand PropertyObjects, such as variables, steps, and module parameters. Use attributes to store additional information, such as the author of a sequence, about a PropertyObject within a sequence file or type palette file.

When you copy a PropertyObject, TestStand adds the attributes stored on the PropertyObject to the new copy. You can create attributes for properties of a type. Attributes of type properties specify the default attributes when you create an instance of a type.

Every PropertyObject includes an `Attributes` property, which is an unstructured container that stores arbitrary data. Although an attribute can be any kind of data, you typically create a name-value pair to give an attribute a name and a data value.

Right-click a PropertyObject on the Variables pane and select **Advanced» Edit Attributes** from the context menu to launch the PropertyObject Attributes dialog box, in which you can create and edit attributes. PropertyObjects with attributes include an **Edit Attributes** button in the Name column of the Variables pane.

Steps with attributes include the word `Attributes` in the step description, and the General panel of the Properties tab on the Step Settings pane includes an **Edit Attributes** button.

Right-click a parameter on the Module tab of the Step Settings pane in the sequence editor or in the Specify Module dialog box in user interfaces and select **Edit Attributes** from the context menu to launch the parameter Attributes dialog box. Parameters with attributes include an **Edit Attributes** button in the Name column of the Parameters Table. You can edit attributes for fields of containers and elements of arrays. TestStand associates parameter attributes with the module parameter, which TestStand stores with the step configuration information, not with the parameter value that TestStand passes to the module. TestStand stores attributes specified in module parameters with the parameter definition for the module, and not with a parameter value passed to the code module.

The TestStand Differ application detects and merges attribute differences. Right-click an item and select **Edit Attributes** from the context menu or click the **Edit Attributes** button in the Name column of the Differ window to launch the related Attributes dialog box. When you right-click an item and select **Replace Item Values in File <*filename*>** from the context menu, TestStand now also copies attribute values. Refer to the *NI TestStand Help* for more information about the TestStand Differ application.

Attributes must have unique names. You can use a container under an `Attributes` property to serve as a namespace and to ensure uniqueness of attribute names. National Instruments recommends using company prefixes as namespaces. For example, the `Attributes` property can include a top-level `NI` container that acts as a namespace for attributes of built-in features. The `NI` attribute can include a `TestStand` attribute for TestStand-related attributes. The `NI.TestStand` attribute includes subproperties for individual TestStand features to prevent conflicts among TestStand features. Similarly, `NI.ATML` would include ATML-related attributes.

Note The report generation and database logging features in TestStand do not support including attribute information in reports or databases. In addition, the Find and Find in Files features in the sequence editor do not support searching attributes for matching attribute names or values. Also, the `PropertyObject.GetXML` and the `PropertyObject.SetXML` methods do not support attribute information in the generated XML.

Refer to the *NI TestStand Help* for more information about the options for editing attributes in the TestStand Sequence Editor or in a TestStand User Interface.

Setting the Policy for Marking Files as Modified

Use the **Do Not Mark Files as Modified if Modification is Caused Only by Automatic TestStand Version Upgrade** option on the File tab of the Station Options dialog box to configure when TestStand marks files as modified. Enable this option if you do not want TestStand to mark files as modified when the only changes to the file are the result of the automatic conversions TestStand makes when you open files from an older version of TestStand in a newer version of TestStand. These automatic conversions occur when TestStand upgrades older versions of built-in, National Instruments-created types and step types and then resolves data format changes that exist between the versions.

This option is disabled by default. When you enable this option, TestStand marks as modified only the files with user-defined changes so you can avoid resaving and revalidating files you did not directly modify. Enabling or disabling this option does not change how TestStand loads the file into memory or saves the file to disk.

This option applies only to National Instruments-created types, step types, and file modifications. Use this option to prevent TestStand from prompting to resave upgraded files, which might require revalidation. When you enable this option, TestStand still marks files as modified when type

conflict resolution results in updates to user-defined types and step types in the file. Resave files in the version of TestStand you are using to optimize performance when opening files.

Regardless of the value of this option, opening files from a TestStand version older than the version you are using requires TestStand to perform an in-memory conversion. Resave files in the version of TestStand you are using to optimize file loading performance.

Refer to the *NI TestStand Help* for more information about the options on the File tab of the Station Options dialog box.

API Additions

Refer to the *TestStand 4.2 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 Although some components of TestStand, such as the TestStand Sequence Editor, expose .NET assemblies, TestStand does not support the use of undocumented assembly API.

Other TestStand Enhancements

TestStand 4.2 includes the following additional enhancements:

- Use the Set Thread Externally Suspended VI within a code module or step type to enable a calling execution to break while the code module or step type performs a lengthy operation or waits an indeterminate period of time. TestStand considers the calling thread to be suspended if the user requests that the execution break. Refer to the *NI TestStand Help* for more information about the `Thread.ExternallySuspended` property. Refer to the *NI TestStand VIs and Functions Help* for more information about the Set Thread Externally Suspended VI.
- Several process model changes support the new reporting features.
- The TestStand Version Selector no longer requires administrator privileges. You can now change the active version of TestStand while logged in as a user with standard privileges.

- The TestStand Version Selector includes the following new command-line options:
  - `/noprompt`—Suppresses prompts while performing activation and launch operations.
  - `/installing`—Displays a status prompt while activating the specified version without displaying errors. You must use this option with the `/version` and `/noprompt` options.

  Click the **Help** button in the TestStand Version Selector to launch the *TestStand Version Selector Help*, which includes more information about available command-line options.
- By default, the sequence editor now allows you to edit read-only files.
- You can now select **Start»All Programs»National Instruments»TestStand *x.x*»Tools** and select the stand-alone tool you want to run outside of the sequence editor, including the Database Viewer, TestStand Deployment Utility, Sequence File Converter, Sequence File Differ, and XML Packaging Utility.
- You can now use additional results to log output parameters without a variable specified to store the result. TestStand 4.1 did not log an additional result for an output parameter not stored in a variable.
- The *NI TestStand Help* and manuals includes 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.

- Chapter 1, *Introduction to TestStand*, of the *Using TestStand* manual now contains a *Creating a TestStand System* section that includes information about creating a TestStand-based test solution, including phases of the development life cycle and the licensing options available for each phase.

National Instruments, NI, ni.com, NI TestStand, 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/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`.

© 1999–2009 National Instruments Corporation. All rights reserved.