

RELEASE NOTES

TestStand™

Version 3.0

These release notes contain system requirements for TestStand 3.0, as well as information about new features, documentation resources, and other changes since TestStand 2.0.1.

Contents

Minimum System Requirements.....	2
Installing Multiple Versions of TestStand on the Same Computer	2
Migrating User Components.....	3
Migrating Your Changes to TestStand Components	4
Known Issues	5
What's New in TestStand 3.0	5
Improved Language and ADE Integration.....	5
LabVIEW Adapter.....	5
LabWindows/CVI Adapter.....	6
C/C++ DLL Adapter.....	6
.NET Adapter.....	6
ActiveX/COM Adapter.....	6
Sequence Adapter	6
Java Step Type.....	6
Simple Operator Interface Development	7
TestStand API Enhancements.....	7
New System Deployment Tools	8
Improved Report Generation and Database Logging	
Capabilities	8
Productivity Enhancements	8
Miscellaneous Features.....	9
Using Microsoft Visual Studio .NET to Debug a Code Module	
Within a TestStand Process.....	9
Documentation	11

Minimum System Requirements

To run TestStand 3.0, National Instruments recommends that your system meet the following requirements:

- Windows 2000/XP/98 or Windows NT 4.0 Service Pack 6a or later (the TestStand User Interface Controls are not supported in Windows 98)
- 400 MHz Pentium class microprocessor (266 MHz minimum)
- 128 MB of memory (64 MB minimum)
- 500 MB of free hard disk space (100 MB minimum)
- SVGA resolution or higher video adapter, with a minimum 800×600 video resolution for small fonts or a minimum 1024×768 for large fonts
- Microsoft Internet Explorer version 6.0 or later (5.5 or later required)

TestStand 3.0 is compatible with the following National Instruments application development environments (ADEs):

- LabVIEW 6.1 or later (LabVIEW 7.0 or later is required to use the TestStand User Interface Controls)
- LabWindows[™]/CVI[™] 6.0 or later (LabWindows/CVI 7.0 or later is recommended for use with the TestStand User Interface Controls)
- Measurement Studio 7.0 (or later) Enterprise Edition is required for integration with Microsoft Visual Studio .NET 2003 or later

Installing Multiple Versions of TestStand on the Same Computer

You cannot install TestStand 3.0 over a previous version of TestStand. In order to install TestStand into a directory that contains a previous version, you must first uninstall the previous version. However, since the uninstallers for previous versions of TestStand remove the <TestStand>\OperatorInterfaces\User directory, you must use the following procedure to safely uninstall the previous version of TestStand and preserve all configuration files and files located in the User subdirectories:

1. Move the <TestStand>\OperatorInterfaces\User directory to a location on your computer that is outside the <TestStand> directory.
2. Run the TestStand Uninstaller by selecting **Start»Control Panel»Add/Remove Programs**.

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 3.0 into this directory.

Although TestStand 3.0 will install on a machine that contains a previous TestStand version, only one version of TestStand can be active at a time. If you must install TestStand 3.0 on the same machine as an earlier TestStand version, you can use the TestStand Version Selector to specify the active version of TestStand. To launch the TestStand Version Selector, select **Start»Programs»National Instruments»TestStand 3.0»TestStand Version Selector**. The TestStand Version Selector application, `TSVerSelect.exe`, is located in the following directory: `C:\Program Files\National Instruments\Shared\TestStand Version Selector`.

If you activate TestStand 3.0 and run an operator interface from your TestStand 2.0.x installation, the operator interface uses the engine, step types, and components from TestStand 3.0. If you activate TestStand 2.0.x and run a TestStand 3.0 Operator Interface or the TestStand Sequence Editor, those applications will not function correctly.

Migrating User Components

You can copy the following directories and files from TestStand 2.0.x to the TestStand 3.0 installation directory:

- `<TestStand>\Components\User`
- `<TestStand>\CodeTemplates\User`
- `<TestStand>\OperatorInterfaces\User`
- `<TestStand>\Cfg\TypePalettes\MyTypes.ini`
- `<TestStand>\Cfg\TypePalettes\SeqEdit.ini`
- `<TestStand>\Cfg\TypePalettes\TestExec.ini`
- `<TestStand>\Cfg\TypePalettes\TestStandDatabaseOptions.ini`
- `<TestStand>\Cfg\TypePalettes\TestStandModelModelOptions.ini`
- `<TestStand>\Cfg\TypePalettes\TestStandModelReportOptions.ini`
- `<TestStand>\Cfg\TypePalettes\Users.ini`

If you have custom Tools menu items in TestStand 2.0.x, use the following procedure to export the items from TestStand 2.0.x and then import them into TestStand 3.0:

1. In TestStand 2.0.x, launch the Customize Tools Menu dialog box from the TestStand Sequence Editor and click **Export Items to File** to launch the Export Tools Menu dialog box.
2. In the Export Tools Menu dialog box, select the menu items to export to a Tools menu file.
3. Create the following directory under TestStand 3.0:
`<TestStand>\Setup\ToolMenusToInstall.`
4. Place the Tools menu file in the new directory.
5. Launch the TestStand 3.0 Sequence Editor to allow TestStand to add the new menu items to the Tools menu and then delete the Tools menu file.

Migrating Your Changes to TestStand Components

TestStand includes several components that you can customize, such as the process models, operator interfaces, and certain step types. If you have made changes to one of these components and placed the changes in the appropriate user directory, TestStand will not overwrite your changes when you install TestStand 3.0. Your modified component will continue to function correctly with TestStand.

To keep the changes you have made to a component and incorporate the new functionality provided in TestStand 3.0, select one of the following options:

- **If you have made substantial or complex changes to the component**—Use a file-differencing tool to determine the changes between the TestStand 3.0 version of the component and the original version of the component that you modified. Then, apply the TestStand 3.0 improvements to your version of the component.
- **If you have made minor changes to the component**—Use a file-differencing tool to determine the changes you made to the component. Reapply your improvements to a copy of the TestStand 3.0 version of the component.

The following types of file-differencing tools are available:

- To compare sequence files, use the TestStand Differ, which you access from the TestStand 3.0 Sequence Editor by selecting **Edit> Diff Sequence File With**.
- To compare text files, use a source code differencing tool such as Microsoft Windiff, or the Diff command, which is located in the Edit menu of the Source and Execution Window in LabWindows/CVI.

- To compare VI files, use the Compare VIs tool in LabVIEW by selecting **Tools»Compare**. This tool is only available if you have the LabVIEW Professional Development System installed.



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 3.0 process models with the corresponding sequence from the TestStand 2.0.x process models without making further modifications. If you customized the TestStand 2.0.x process models, you must ensure that all subordinate components used by the process models are located and that any of those components that are ActiveX servers are registered. The main process model sequence for TestStand 2.0.x uses separate sequences, DLLs, and ActiveX servers to support database logging and report generation features.

Known Issues

If you use the TestStand Step Into command to launch Microsoft Visual Studio .NET to debug a code module from the C/C++ DLL Adapter or the .NET Adapter, you must first disable script debugging in Microsoft Internet Explorer. To do so, select **Tools»Internet Options** from within Internet Explorer and then click the **Advanced** tab. Then, under the **Browsing** category, select the **Disable script debugging** option. Failure to disable script debugging can cause Visual Studio .NET to hang while attaching to the TestStand process.

What's New in TestStand 3.0

This section describes the new features in TestStand 3.0 and other changes since TestStand 2.0.1.

Improved Language and ADE Integration

LabVIEW Adapter

The LabVIEW Adapter (formerly the LabVIEW Standard Prototype Adapter) offers the following new features when used with LabVIEW 7.0:

- Call VIs with any connector pane
- Run VIs with the LabVIEW Run-Time Engine
- Call VIs on remote systems, including LabVIEW Real-Time (RT) systems
- Significant performance improvements

LabWindows/CVI Adapter

TestStand 3.0 allows you to use the LabWindows/CVI Adapter (formerly the C/CVI Standard Prototype Adapter) to call LabWindows/CVI functions with any prototype. Parameter data types include numerics, strings, objects, arrays, and C structures.

C/C++ DLL Adapter

In TestStand 3.0, you can use the C/C++ DLL Adapter (formerly the DLL Flexible Prototype Adapter) to perform the following tasks:

- Call static global and class methods in C++ DLLs
- Integrate with Microsoft Visual Studio .NET 2003 for code generation, editing, and debugging
- Pass TestStand data to Measurement Studio C++ class object parameters, such as CNiReal64Vector and CNiReal64Matrix

.NET Adapter

Use the .NET Adapter to perform the following tasks:

- Call methods and access properties in .NET assemblies
- Integrate with Microsoft Visual Studio .NET 2003 for code generation, editing, and debugging

ActiveX/COM Adapter

In TestStand 3.0, you can use the ActiveX/COM Adapter (formerly the ActiveX Automation Adapter) to call COM servers with custom interfaces. This feature is limited to automation data types.

Sequence Adapter

In TestStand 3.0, you can use the Sequence Adapter to launch new threads and executions that are initialized as single-threaded apartment, which allows the thread to create or launch dialog boxes that contain ActiveX controls. In addition, the TestStand Engine launches Edit <Step Type> dialog boxes for steps in threads that are initialized as single-threaded apartment.

Java Step Type

TestStand 3.0 includes an example step type for calling Java class methods.

Simple Operator Interface Development

TestStand 3.0 introduces the TestStand User Interface (UI) Controls, which are ActiveX controls designed to radically simplify the development of custom operator interfaces.

Example operator interfaces are available for LabVIEW, LabWindows/CVI, Microsoft Visual Basic .NET, C#, and C++ (MFC).

TestStand API Enhancements

The following features are some of the many enhancements to the TestStand API for TestStand 3.0:

- Addition of a dual ActiveX interface
- Ability to programatically configure the following features:
 - Adapters
 - Step code modules and parameters
 - Search directories
 - External viewers
 - Tools menu



Note TestStand 3.0 rearranged many of the adapter-specific properties of steps, which are located under the TS.SData property of a step. Do not access these properties directly. Instead, use the adapter-specific module interface for the Module object for the step.

- A new Thread.RunTimeVariables property, which allows you to attach object references to adhere to the lifetime of a thread
- A new SequenceContext.Calling step property, which allows you to see which step calls an asynchronous sequence
- A new AddPostStepCustomUIMessage method, which allows you to register a post-step expression that generates an event when True
- Support in the EditArgs class for the SelectedPropertyObjects property, which allows a Tools menu item to edit a selected variable, such as arrays
- A new UIMessage.Acknowledge method that allows an operator interface to respond to a UI message before returning from an event invocation
- Model State events which communicate UUT status information to operator interfaces
- Complete methods and properties in the StepType and Step classes
- Definitions for all station options in the new StationOptions class

New System Deployment Tools

TestStand 3.0 introduces the TestStand Deployment Utility, which creates an installer and an installable image for deploying your TestStand system to one or more target machines.

The installer you create can include the following:

- Your TestStand workspace and associated files
- Any files stored in your TestStand `User` directories
- The TestStand Engine

Improved Report Generation and Database Logging Capabilities

TestStand 3.0 offers report generation features such as built-in XML reporting and on-the-fly report generation. TestStand 3.0 also offers on-the-fly database logging.

Productivity Enhancements

The following features represent some of the major productivity enhancements in TestStand 3.0:

- Find and Replace capability
- Next/Previous buttons in the Sequence File windows in the TestStand Sequence Editor
- Ability to change the step type and adapter for an existing step
- Improved layout and editing of parameters on Specify Module dialog boxes
- New execution debugging options that break on step failure as well as on sequence failure
- Ability for the TestStand Sequence Editor to automatically assign unique IDs to steps and for Goto steps to use these unique IDs as targets
- Links to the *TestStand Help* from within all TestStand dialog boxes
- New station option that allows branching during an interactive execution
- New station option that determines whether failures and errors from nested executions propagate to the calling execution
- Enhancements to breakpoints and watch expressions, including the following:
 - Save breakpoints and watch expressions with your workspace
 - Disable breakpoints
 - Break on watch expression change

- Smart editing in expression controls
- Integration of help into Microsoft Visual Studio .NET 2003
- VI and function reference help

Miscellaneous Features

- License management
- Specify Module dialog boxes and Edit <Step> dialog boxes support read-only mode
- Remote Engine now displays an icon in the system tray. Use this system tray icon to get engine status information and to exit the Remote Engine application
- Updates to the LabVIEW palette for TestStand provide improved usability, as well as new polymorphic VIs for getting and setting property values
- Removed the TestStand Sequence Editor option to allow editing of National Instruments step types
- Improvements to the Multiple Numeric Limit Test step type
- New switching expression functions, which include Find Route capability
- Basic math and string functions in expressions

Using Microsoft Visual Studio .NET to Debug a Code Module Within a TestStand Process

If you use Microsoft Visual Studio .NET to launch or attach to a TestStand process for debugging a code module called by the C/C++ DLL Adapter or the .NET Adapter, you must ensure that the Visual Studio project for the code module allows you to debug unmanaged code. If you do not do this, the Step Into command from TestStand will not work.

In Visual Studio, use the Project Properties dialog box to specify which debugger to use. Select **Project»Properties** in Visual Studio to launch the Project Properties dialog box. Since the options for enabling debuggers vary according to the type of project, use Table 1 as a guideline for each project type.

Table 1. Recommended Debugger Settings by Project Type

Project Type	Recommended Debugger Setting
Visual Basic .NET	Visual Basic projects automatically allow you to debug managed code. To allow TestStand to step into a code module, enable the Unmanaged Code Debugging option in the Configuration Properties group of the Project Properties dialog box.
C# .NET	C# projects automatically allow you to debug managed code. To allow TestStand to step into a code module, enable the Enable Unmanaged Debugging option in the Configuration Properties group of the Project Properties dialog box.
C++ .NET	<p>C++ .NET projects can contain both managed and unmanaged code.</p> <p>The Debugger Type option in the Configuration Properties group of the Project Properties dialog box specifies one of the following options:</p> <ul style="list-style-type: none">• Native—Debugs unmanaged C++ code only.• Managed Only—Debugs managed code that runs under the common language runtime.• Mixed—Invokes debuggers for both managed and unmanaged code. Use this setting to allow TestStand to step into a code module.• Auto—Attempts to determine the debugger type based on compiler and EXE information. This is the default setting. <p>Note In some instances, the Auto option incorrectly selects the managed code debugger instead of selecting both managed and unmanaged.</p>
C++ (Native)	Native C++ projects, such as Win32 projects, already allow you to debug unmanaged code.

Documentation

TestStand 3.0 includes a completely redesigned documentation set featuring printed and electronic resources for your reference.

The following printed resources are available:

- *TestStand Quick Start Guide*
- *TestStand System and Architecture Overview Card*
- *Using TestStand*—Contains information about the TestStand environment and the basic features you use to build and run sequences. For information about using TestStand with specific National Instruments ADEs, refer to the following manuals:
 - *Using LabVIEW with TestStand*
 - *Using LabWindows/CVI with TestStand*
- *TestStand Reference Manual*—Contains information about TestStand concepts and features.
- *TestStand User Interface Controls Reference Poster*
- *TestStand API Reference Poster*

The following electronic resources are also available:

- *TestStand Bookshelf*—Use the *TestStand Bookshelf* to access all of the documentation in electronic format. You can also search the *TestStand Bookshelf* by keyword and/or phrase to find information quickly. To access the *TestStand Bookshelf*, select **Start»Programs»National Instruments»TestStand 3.0»Online Help»TestStand Bookshelf**.
- *TestStand Help*—Contains information about the TestStand environment and the TestStand UI Controls and Engine APIs. The *TestStand Help* also includes basic information about using an ActiveX automation server. To access the *TestStand Help*, select **Start»Programs»National Instruments»TestStand 3.0»Online Help»TestStand Help**.



Note If you are opening the *TestStand Help* from the <TestStand>/Doc/Help directory, National Instruments recommends that you open the TSHelp.chm file. The TSHelp.chm file is a collection of all of the help files available in TestStand and provides a complete table of contents and index.

CVI™, LabVIEW™, Measurement Studio™, National Instruments™, NI™, ni.com™, and TestStand™ are trademarks of National Instruments Corporation. Product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

© 1999–2003 National Instruments Corp. All rights reserved. Printed in Ireland.



322519F-01

Jul03