

OVERVIEW CARD

NI TestStand™ System and Architecture

NI TestStand is flexible test management software that offers the following major features:

- Out-of-the-box configuration and components that provide a ready-to-run, full-featured test management environment.
- Numerous methods for modifying, configuring, and adding new components, which provide extensibility so you can create a test executive that meets specific requirements without altering the core TestStand Engine. You can upgrade to newer versions of TestStand without losing the customizations.
- Sophisticated sequencing, execution, and debugging capabilities, and a powerful TestStand Sequence Editor that is separate from the user interfaces.
- User interface controls for creating custom user interfaces and sequence editors. You can also create custom user interfaces in any programming language that can host ActiveX controls or control ActiveX Automation servers.

Refer to the *NI Trademarks and Logo Guidelines* at ni.com/trademarks for more information on 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 Patents Notice* at ni.com/patents. You can find information about end-user license agreements (EULAs) and third-party legal notices in the readme file for your NI product. Refer to the *Export Compliance Information* at ni.com/legal/export-compliance for the National Instruments global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data. **NI MAKES NO EXPRESS OR IMPLIED WARRANTIES AS TO THE ACCURACY OF THE INFORMATION CONTAINED HEREIN AND SHALL NOT BE LIABLE FOR ANY ERRORS.** U.S. Government Customers: The data contained in this manual was developed at private expense and is subject to the applicable limited rights and restricted data rights as set forth in FAR 52.227-14, DFAR 252.227-7014, and DFAR 252.227-7015.

© 2003–2014 National Instruments. All rights reserved. Printed in Ireland.

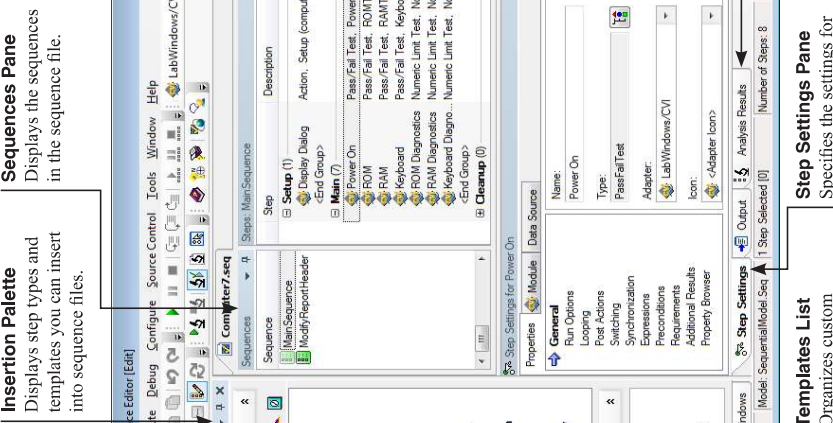
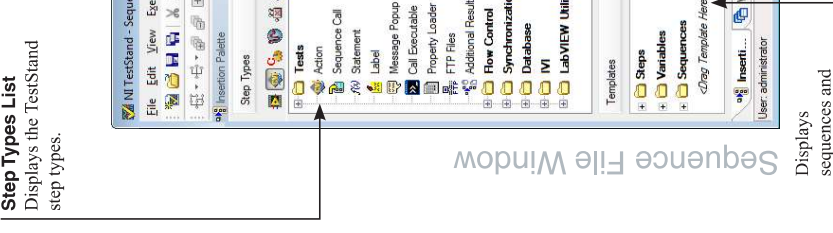
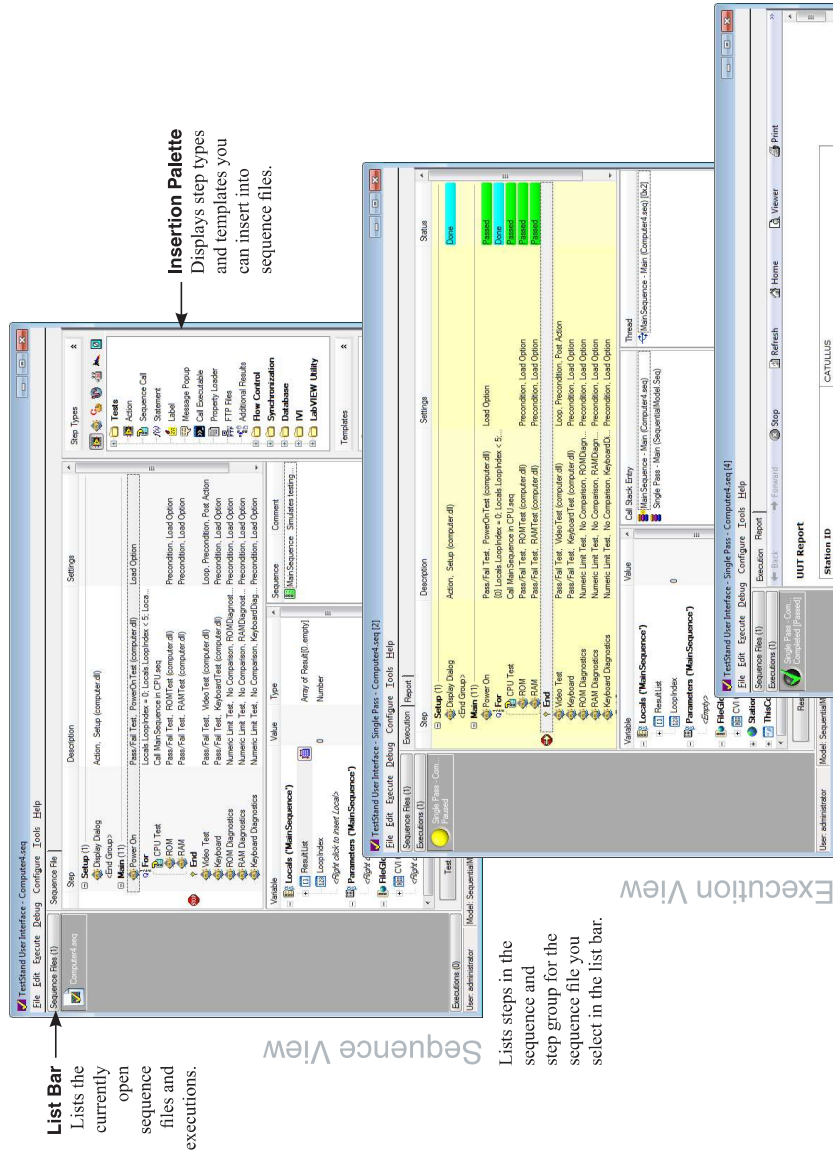
Architecture Overview

- **TestStand Sequence Editor**
TestStand development environment for creating, modifying, executing, and debugging sequences.
- **Custom User Interfaces**
Customizable applications that, depending on mode, edit, execute, and debug test sequences on a test station. User interfaces are available in several different programming languages and include full source code, so you can modify them to meet specific needs.
- **Process Models**
Define standard operations for all test sequences, such as identifying the unit under test (UUT), notifying the operator of pass/fail status, generating a test report, and logging results. TestStand includes three fully customizable process models: Sequential, Parallel, and Batch.
- **User Interface Controls**
Set of ActiveX controls and support APIs for creating custom user interfaces.
- **TestStand Engine**
DLLs that provide an extensive ActiveX Automation API for controlling and interacting with TestStand. The sequence editor, TestStand User Interface (UI) Controls, and user interfaces use this API.
- **Sequence File Executions**
Created by the TestStand Engine when you execute a test sequence using the sequence editor or a user interface.
- **Adapters**
Allow TestStand to call code modules in a variety of different formats and languages. Adapters also allow TestStand to integrate with various ADEs to streamline test code generation and debugging.
- **Code Modules**
Program modules, such as LabVIEW VIs (.vxi) or DLLs, that contain one or more functions to perform a specific test or action. TestStand adapters call code modules.
- **Built-in Step Types**
Define the standard behaviors for common testing operations. Step types use adapters to call code modules that return data to TestStand for further analysis, call executables, launch dialog boxes, implement test execution logic, or perform other operations.
- **User-Defined Step Types**
Define custom step properties and default behaviors for each step of that custom type. You can also create custom data types.
- **Templates**
Create custom sequences, steps, and variables to use as templates to build sequence files.



User Interface Overview

TestStand includes separate user interface applications developed in LabVIEW, LabWindows/CVI, Visual Basic, .NET, C#, and C++ MFC. Because TestStand also includes the source code for each user interface, you can fully customize the user interfaces. You can create custom user interfaces using any programming language that can host ActiveX controls or control ActiveX servers. With the user interfaces in Editor Mode, you can modify sequences and create and modify sequence variables, sequence parameters, step properties, and so on. With the user interfaces in Operator Mode, you can start multiple concurrent executions, set breakpoints, and single-step through sequences.



Displays the threads, call stack, and steps for the execution you select.

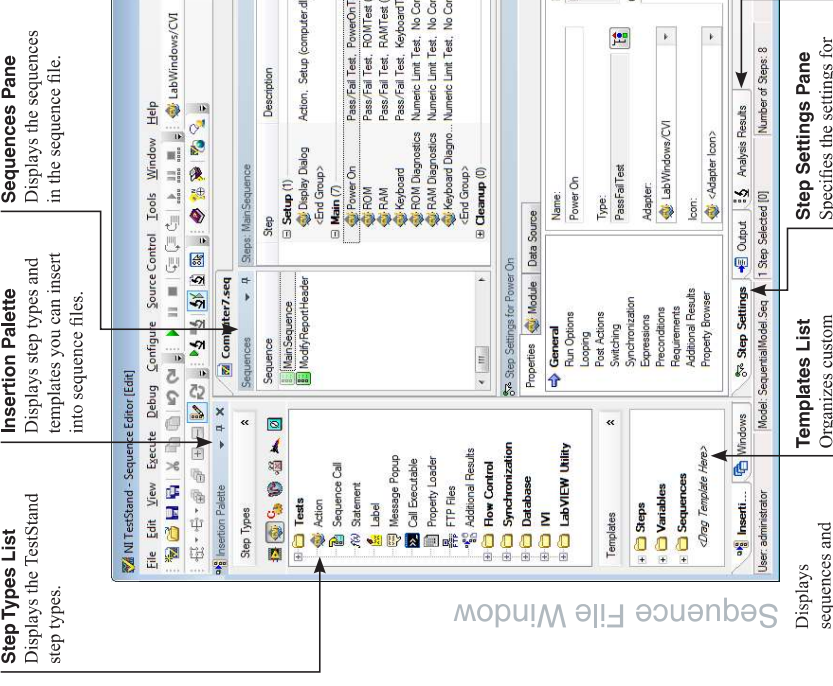
Report View

Displays the report for the execution you select.

Step	Status	Measurement	Units	Low Limit	High Limit	Comparison Type
Display Dialog	Done					
Power On	Passed					
TestResults/Data						
Power Off	True					
For	Done					

TestStand Sequence Editor Overview

You can use the fully customizable TestStand development environment to create, modify, sequence editor to modify step types, data types, process models, and process model plug-in auto-hiding, and floating panes to optimize development tasks.



Workspace Pane

Manages projects for source code control (SCC) integration and deployment. TestStand integrates with third-party SCC packages to add files, obtain the latest versions of files, and check files in and out. Use TestStand project (.tpj) files to organize sequence files and code module files in folders.

