

OVERVIEW CARD

TestStand™ System and Architecture

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

- Out-of-the-box configuration and components that give you a ready-to-run, full-featured test management environment.
- Numerous methods for modifying, configuring, and adding new components. These methods provide extensibility and enable you to create a test executive that meets your particular requirements without altering the core engine. You can also upgrade to newer versions of TestStand without losing your customizations.
- Sophisticated sequencing, execution, and debugging capabilities, and a powerful sequence editor that is separate from the operator interfaces.
- User interface controls for creating custom operator interfaces.
- Example operator interfaces with source code for National Instruments LabVIEW, LabWindows™/CVI™, Microsoft Visual Basic .NET, C#, and C++ (MFC).
- Open language interface that provides support for many application development environments (ADEs). You can create code modules in a wide variety of ADEs and call preexisting modules or executables. You can also create your own operator interface in any programming language that can host ActiveX controls or control ActiveX automation servers.
- Comprehensive application programming interface (API) for building multithreaded test systems and other sophisticated test applications.
- Integration with third-party source code control (SCC) packages.
- Deployment tools to aid in transferring a test system from development to production.

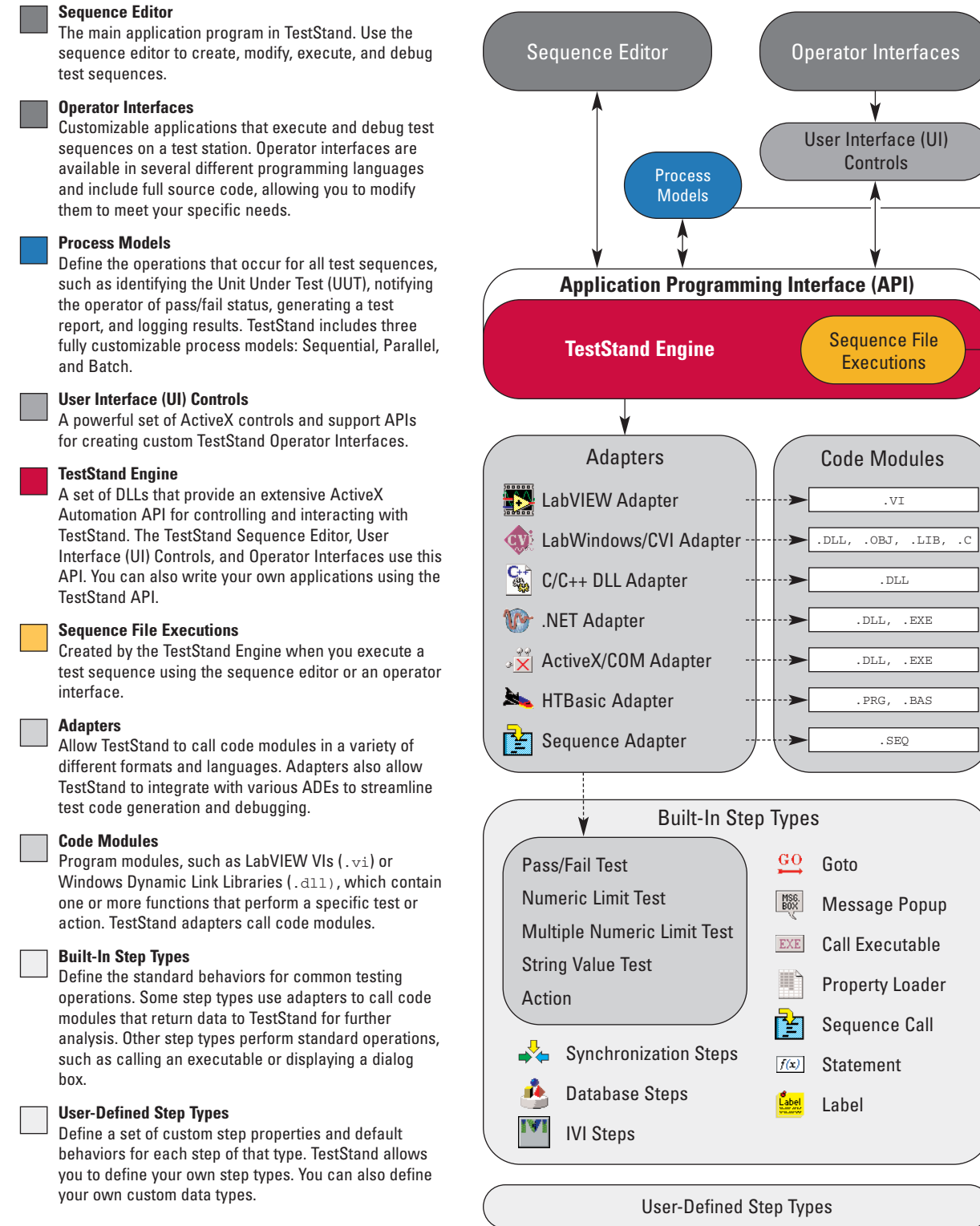


CVI™, LabVIEW™, National Instruments™, NI™, ni.com™, and TestStand™ are trademarks or trade names of National Instruments Corporation. Product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products, refer to the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

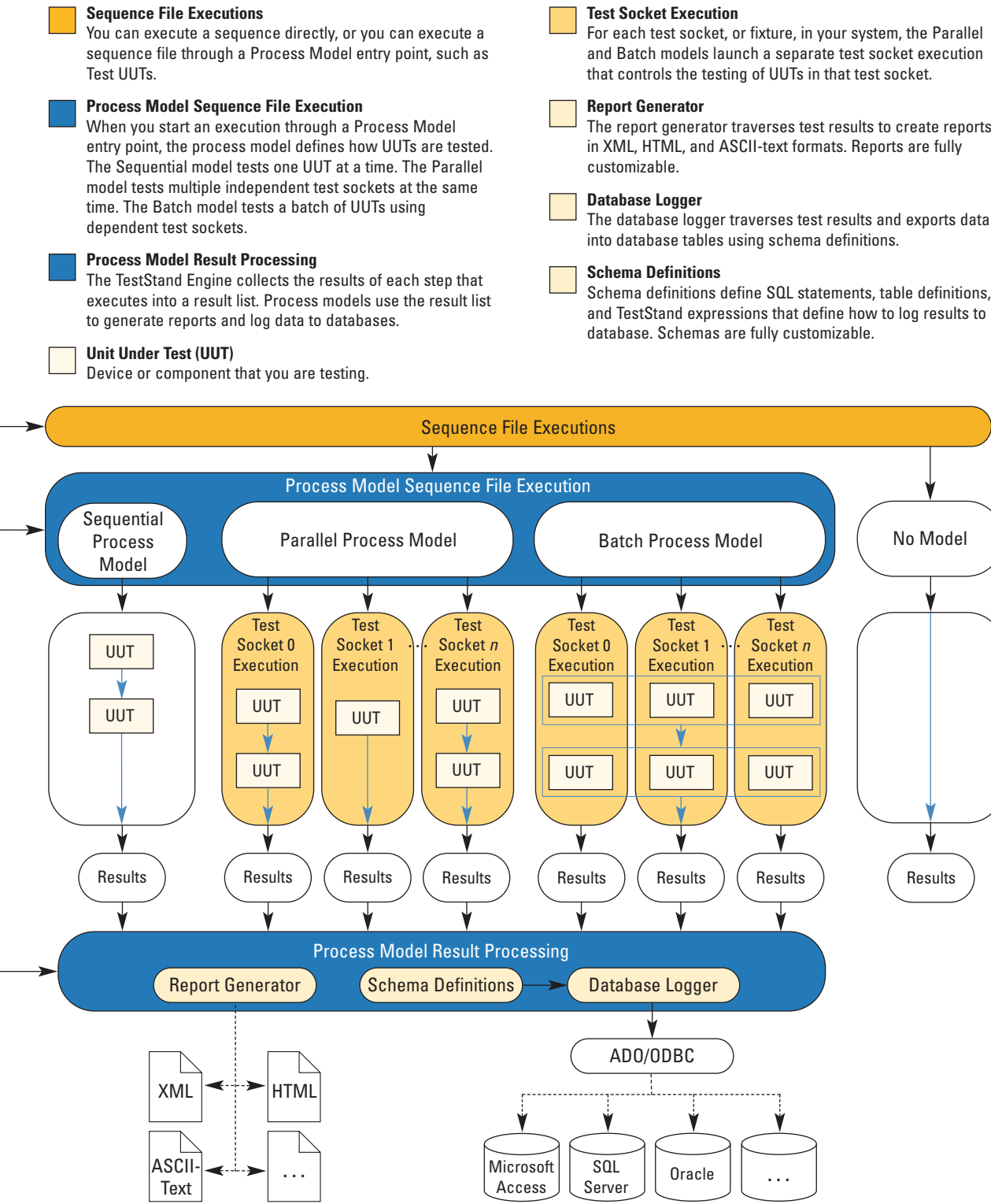
© 2003 National Instruments Corporation. All rights reserved. Printed in Ireland.



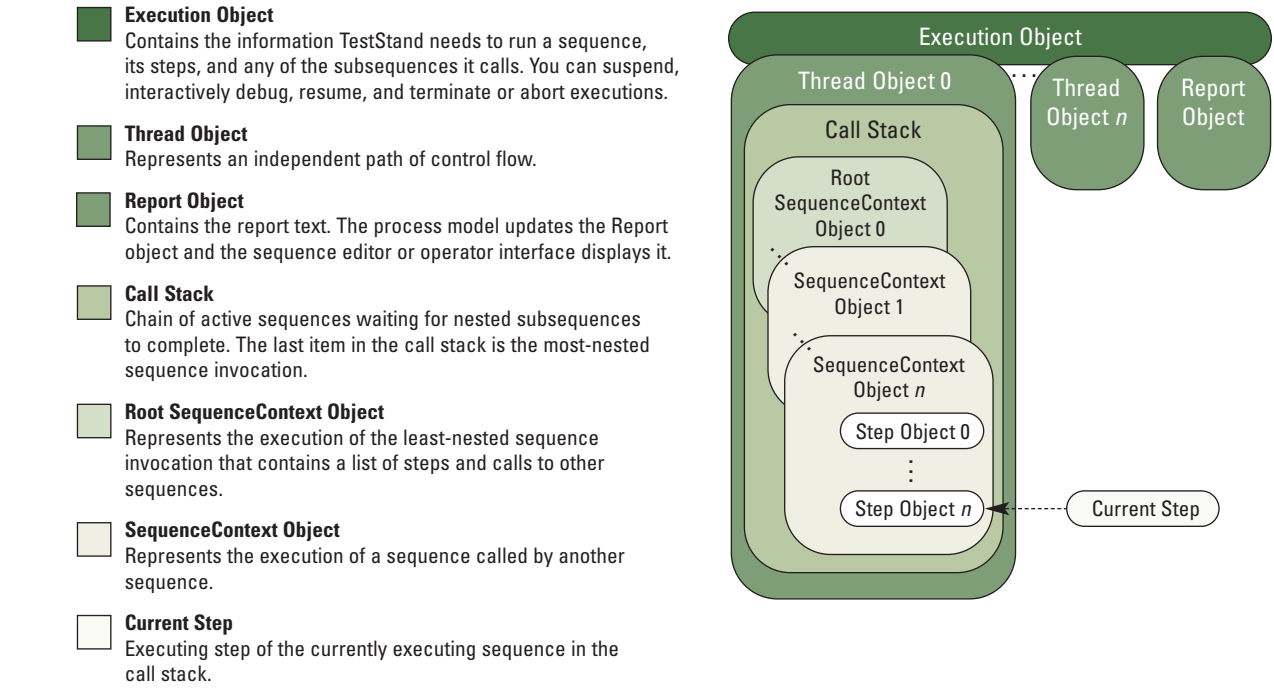
Architecture Overview



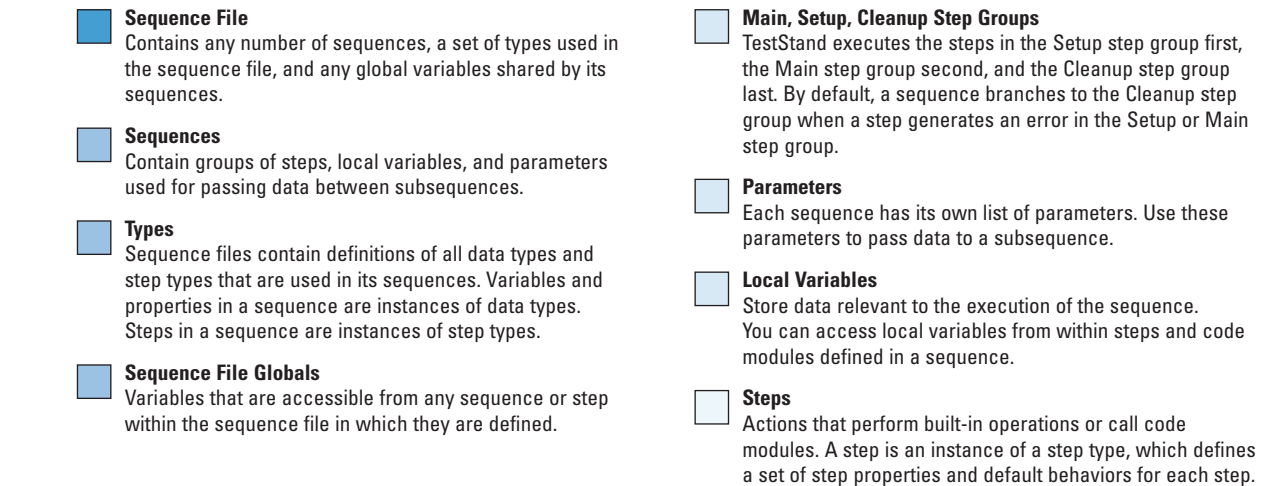
Sequence File Execution Flow



Execution Object Structure



Sequence File Structure



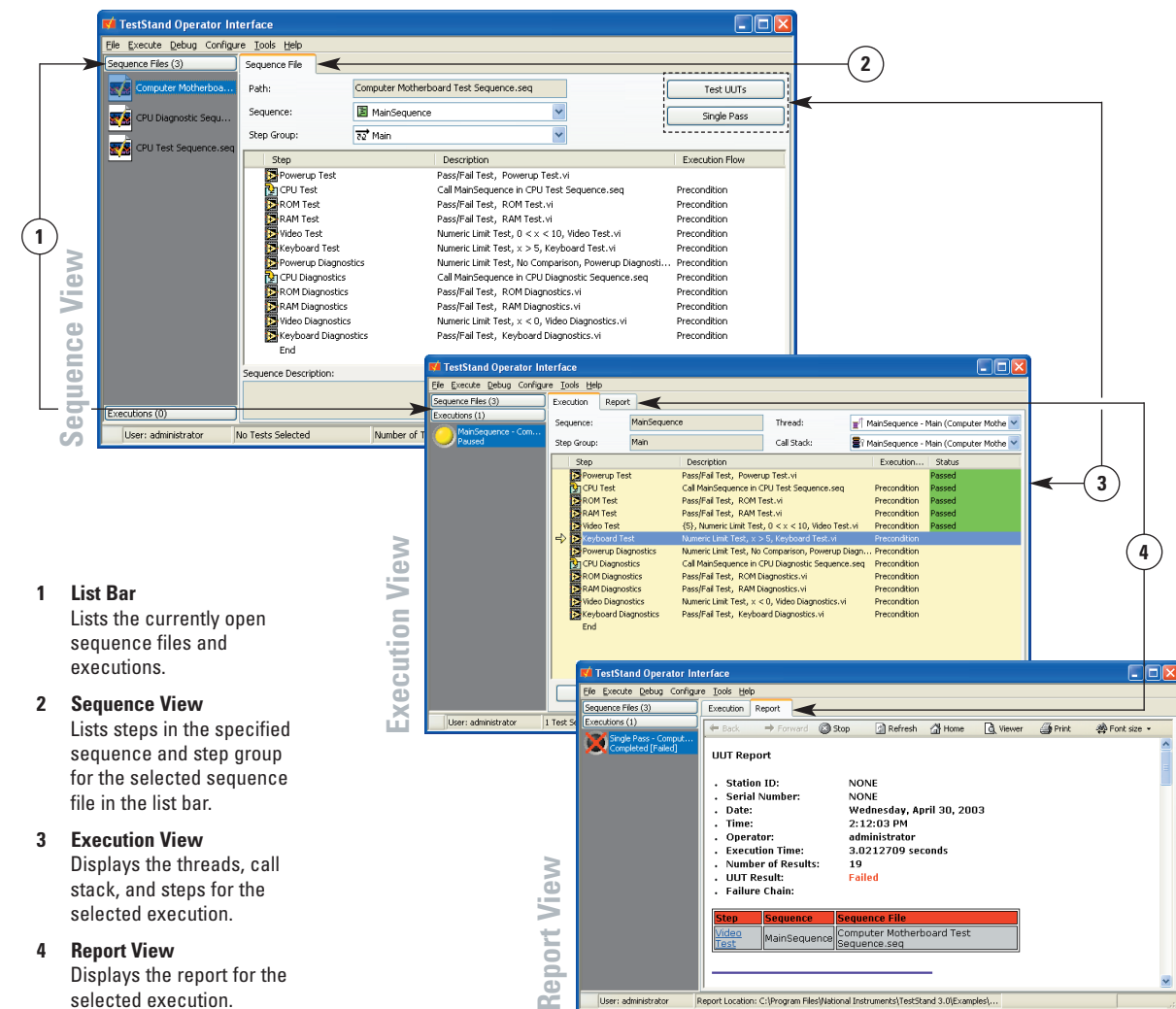
OVERVIEW CARD

TestStand™ System and Architecture

Operator Interface Overview

TestStand includes multiple operator interface examples for out-of-the-box use. Each operator interface is a separate full-featured application program. These interfaces—developed in LabVIEW, LabWindows/CVI, Microsoft Visual Basic .NET, C#, and C++ (MFC)—are available in both source and executable formats. You can also create your own operator interface using any programming language that can host ActiveX controls or control ActiveX automation servers.

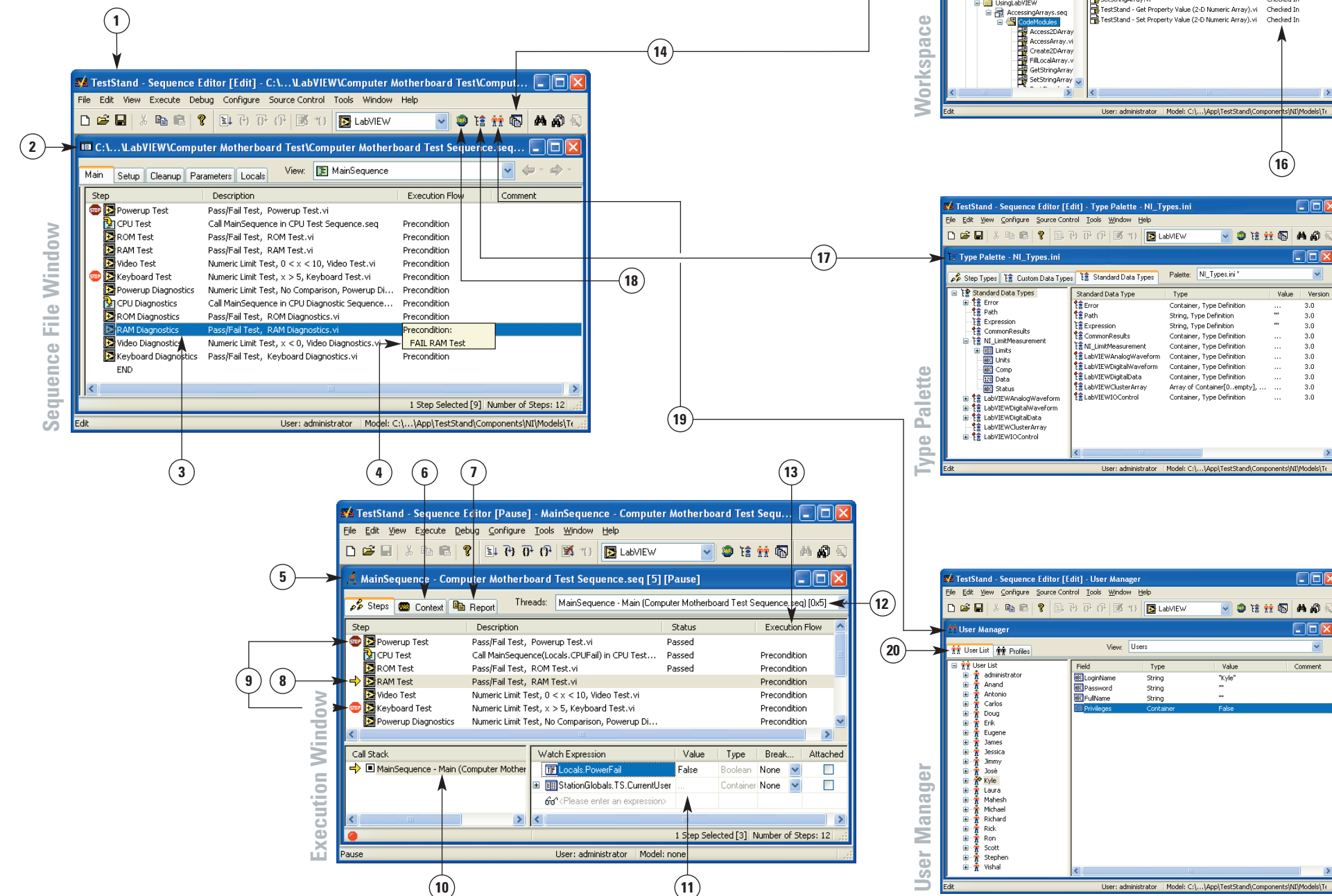
The TestStand Operator Interfaces are fully customizable. Like the TestStand Sequence Editor, the operator interfaces allow you to start multiple concurrent executions, set breakpoints, and single-step through executions. While the operator interfaces do not allow you to modify sequences, display sequence variables, sequence parameters, step properties, and so on, you can add these capabilities to the operator interfaces using the TestStand UI Controls. For more information, refer to the *TestStand User Interface Controls Reference Poster*.



Sequence Editor Overview

The TestStand Sequence Editor is an application program in which you create, modify, execute, and debug test sequences. The sequence editor gives you easy access to all of the powerful TestStand features, such as step types and process models. The sequence editor uses debugging tools that you are familiar with in ADEs such as LabVIEW, LabWindows/CVI, and Microsoft Visual Studio .NET. These include breakpoints, single-stepping, stepping into or over function calls, tracing, a variable display, and the Watch Expression pane.

In the TestStand Sequence Editor, you can start multiple concurrent executions. You can also execute multiple instances of the same sequence or execute different sequences at the same time. Each execution is displayed in a separate Execution window. In trace mode, the Execution window displays the steps in the currently executing sequence. When an execution is suspended, the Execution window displays the next step to execute and provides single-stepping options.



- Sequence Editor**
TestStand development environment for creating, modifying, executing, and debugging sequences.
- Sequence File Window**
Displays sequences and other items found in a sequence file.
- Step**
Performs built-in operations or calls code modules.
- Tool Tip**
Displays details about the item under the mouse cursor.
- Execution Window**
Displays the threads, call stack, and steps that an execution runs.
- Context View**
Displays the run-time variables and properties that steps can access as well as their values.
- Report View**
Displays the report for the current execution.
- Execution Pointer**
Points to the currently executing step at the selected call stack level.
- Breakpoints**
Step settings that suspend an execution before executing the step.
- Call Stack Pane**
Displays the nested sequence invocations for the selected thread.
- Watch Expression Pane**
Monitors the values of specified variables and properties during an execution.
- Thread List**
Contains a list of threads in the current execution.
- Execution Flow**
Indicates the control flow settings for steps, such as preconditions, synchronization, and post actions.
- Workspace**
Manages projects for source code control (SCC) integration and deployment.
- Project**
Organizes sequence files and code module files in folders.
- Source Code Control (SCC)**
TestStand integrates with an SCC provider to add files, obtain the latest versions of files, and check files in and out.
- Type Palette**
Contains definitions of custom data types and step types that all sequence files can use.
- Station Globals**
Displays the variables that can be accessed from anywhere within TestStand and maintained from one session to the next.
- User Manager**
Administers users, login names, passwords, and privileges.
- User List**
Displays users for the test station.

Documentation Map & Overviews

