Evaluating a Test Executive:

Feature Comparison Matrix

National Instruments TestStand is a ready-to-run test management environment for organizing, controlling, and executing your automated prototype, validation, and manufacturing test systems. Quickly build your test sequences with TestStand by incorporating tests written in any programming language. Built on a high-speed, multithreaded execution engine, TestStand delivers the performance to meet your most rigorous test throughput requirements. TestStand is also completely customizable, so you can modify and enhance it to match your specific needs, including customizing the operator interface, generating custom reports, and modifying sequence execution requirements. Using TestStand, you can focus your engineering efforts on more important initiatives, while TestStand manages the common test management tasks for you.

This document is intended to assist you in making the important decision of whether to buy an off-the-shelf test executive or build your test executive software in house. It is also useful for conducting a head-to-head comparison of available commercial test executive software. It highlights features to look for and questions to answer when evaluating test executive software. The document is in the form of a set of tables, each of which addresses a group of test executive or test system features. You can use this matrix both as a tool in your test software evaluation process, and as an overview of some of the more important features of TestStand.

This feature matrix covers the following feature categories:

- Test Execution
- Sequence Development
- Reporting/Result Management
- Customization Features
- Documentation
- Support and Training
- An Open Industry Standard Architecture
- Purchasing Options

Requirement	NI TestStand	Competitive Test Executive or In-House System

Test Execution

Platforms Supported		
Windows 2000	Yes	
Windows XP	Yes	
Windows Vista	Yes	
Macintosh, Linux	No. However, tests developed using NI LabVIEW for Mac, or Linux can be automated remotely using the LabVIEW adapter in NI TestStand for Windows.	
Supported Test Code Module Formats		
LabVIEW Virtual Instruments (.VI)	Yes	
Dynamic Link Libraries (.DLL)	Yes	
.NET Assemblies (.DLL)	Yes	
ActiveX Servers (.DLL and .EXE)	Yes	

Object Files (.OBJ)	Yes	
Static Libraries (.LIB)	Yes	
Standalone Executables (.EXE)	Yes	
HT Basic Files (.PRG and .BAS)	Yes	
Supported Development Environments		
NI LabVIEW	Yes	
NI LabWindows/CVI	Yes	
Visual C++ 6.0	Yes	
Visual C++ .NET	Yes	
Visual Basic 6.0	Yes	
Visual Basic .NET	Yes	
C#	Yes	
Visual J++	Yes	
Borland C++ Builder, Delphi, or any other environment that can generate dynamic link libraries or ActiveX servers	Yes	
Java	Yes. Examples provided with TestStand upon installation.	
ATLAS	Yes. Examples provided with TestStand upon installation.	
PERL	Yes. Examples available online at ni.com/zone.	
TCL	Yes. Examples available online at ni.com/zone.	
HP VEE	Yes. Examples provided with TestStand upon installation.	
HT Basic	Yes. Version 8.0 and above (Windows only).	
Execution		
Does the Test Executive handle multithreaded execution?	Yes. TestStand can easily run any number of executions in parallel using either Batch or Parallel process models. You can independently debug or terminate each execution. Each execution runs in a separate operating system thread.	
Can an execution programmatically launch other executions that run in parallel?	Yes.	
Can test sequences synchronize with each other when they run in separate threads?	Yes. TestStand includes a set of synchronization tools to notify other threads of their requirements.	
How fast is the Test Executive?	Execution speed depends on the types of code modules you call and the number of options you enable. To help you determine the execution speed you can expect in your application, TestStand ships with a set of benchmark sequences.	

Sequence Files		
Can contain multiple test sequences?	Yes	
Can contain sequences that automatically run when the sequence file loads and unloads?	Yes	
Can contain variables shared by multiple sequences?	Yes	
Sequence Flow Control		
Pre-Conditions?	Test steps can specify the conditions that must be true before they can execute. These conditions can be based on previous step results or the values of variables and step properties.	
Post Actions?	Test steps can specify an action to take based on their results or any other condition. Actions include branching, setting breakpoints, terminating, or calling a subsequence.	
Looping?	Test modules can be configured to loop based on any condition.	
Flow control?	Branching and looping can be implemented with common flow control tools such as If statements, While loops, For loops, and Case statements	
Variables		
Test Sequences can define variables which share information between test steps?	You can define any number of variables at the sequence, sequence file, and test station levels. Variables can be numbers, booleans, strings, arrays, or user-defined objects and can be shared across threads.	
Variables can be accessed from test code modules?	All TestStand variables and properties can be accessed within any test code module regardless of the language the module is developed in. You can also pass variables directly to DLL functions, ActiveX server methods, and sub-sequences.	
Debugging		
Can set sequence breakpoints?	Yes	
Can break based on test results?	Yes	
Can step into?	Yes	
Can step out of?	Yes	
Can step over?	Yes	
Can interactively set next	Yes	

step?		
Can run or loop selected tests?	Yes	
Includes a variable browser/viewer and watch expressions?	Yes	
Supports conditional breakpoints and watch expressions?	Yes	
Can break, resume, or terminate a selected sequence execution or all active executions?	Yes	
Can step into CVI DLL's and source files?	Yes	
Can step into LabVIEW VI's?	Yes	
Can step into Visual Studio .NET languages?	Yes, with the addition of NI Measurement Studio	
Remote/Distributed Execution		
Can test sequences call sub-sequences on remote machines?	Yes. TestStand uses Microsoft DCOM to support remote execution of sequences. You can call any number of sequences on any number of remote machines.	
Can parameter values be passed to and returned from remotely executed sequences?	Yes	
Can execute LabVIEW VIs remotely on other networked machines running only LabVIEW?	Yes. Through the NI TestStand LabVIEW adapter.	
Test Limits		
Can test limits be stored and edited in the test steps they apply to?	Yes. This is the default if you do not use external limit files.	
Can test limits be imported and exported to external files?	Yes	
What limit file formats are supported?	NI TestStand can import, export, and view comma and tab separated text limit files, and Microsoft Excel spreadsheet limit files.	
Can test limits be dynamically loaded from external files at run-time?	Yes	
Can test limits be imported and exported to databases?	Yes	
Can test limits be dynamically loaded from a database at run-time?	Yes	
Error Handling		
Hardware and software initialization and cleanup steps are separated from test steps?	Each sequence has a Setup and Cleanup step list to handle initialization and cleanup. Each	

	sequence also has a Main step list where you place the steps that test your UUT.	
Test sequences automatically cleanup properly if an error occurs?	Even if an error occurs, NI TestStand runs the cleanup steps for the active sequences to return the system to a known state.	
Access violations and other fatal system errors in test code modules are trapped and converted into step execution errors?	Yes	
Test steps can be configured to ignore errors?	Yes	
Ability to break on errors for debugging?	You can enable the break-on-run-time-error feature to assist in your sequence debugging.	

Sequence Development

Sequence Editing		
Does the sequence editor support standard Windows editing features?	Yes. The NI TestStand sequence editor supports drag and drop editing, right click context menus, cut/copy/paste, multiple item selections, undo/redo, and toolbars.	
Can you view multiple files simultaneously and drag and drop steps and variables between files?	Yes. The NI TestStand sequence editor presents a standard multiple document interface similar to applications such as Word and Excel.	
Can you add comments to sequences and sequence steps?	Yes. You can place comments on sequence files, sequences, steps, variables, and properties.	
Can you easily browse subsequences and other views within the sequence development environment?	Yes. The next and previous sequence view buttons simplify code reviews and increase useability.	
Can you add requirements coverage information to sequences and sequence steps?	Yes. You can document requirement coverage information in steps and sequences.	
Types of Sequence Steps		
What types of steps can be used in a test sequence?	NI TestStand allows you to define your own types of test steps. However, NI TestStand comes with a set of pre-defined step types that are sufficient for many tasks. These step types are: ● **Action** - Calls a test module that performs an action and optionally returns error information. ● **Numeric Limit Test** - Calls a test module that returns a numeric measurement, and compares the measurement against the limits you	

	specify. ● **Multiple Numeric Limit Test** – Calls a test module that returns a user-defined number of numeric measurements, and compares the measurements against unique limits for each measurement or against a defined set of limits for all measurements. ● **String Value Test** - Calls a test module that return a string value, and compares the string value against the expected string value you specify. ● **Pass/Fail Test** - Calls test modules that return a boolean pass or fail result. ● **Sequence Call** - Calls other sequences. You can pass values to and receive values from the sequences you call. ● **Statement** - Performs calculations with variables. ● **Message Popup** - Displays operator prompts. ● **Call Executable** - Calls an external executable. ● **Limit Loader** - Loads test limits from an external file. ● **Flow Control** – A set of steps to facilitate the control of the sequence's execution order. ● **Database** - A set of steps for low level manipulation of data and databases. ● **Synchronization** - A set of steps to facilitate communication and control between application threads. ● **IVI** - Instrument control steps which expose full functionality of standard instruments.	
Reusable Test Sequences		
Any number of parent sequences can call a test sequence?	Yes	
Sequences can accept and return any number of parameters?	Yes	
Sequences can be called reentrantly (required for multithread safety) and recursively?	Yes	
A sequence can dynamically select which subsequence to	Yes	

call at runtime?		
User Management		
User Login and Logout?	Yes	
Optional User Privilege Enforcement?	Yes	
User Group Editor?	Yes, user groups provided for Operator, Technician, Developer, and Administrator. You can modify these groups or define your own.	
Ability to automatically use Windows system/domain login?	Yes	
Ability to define new privileges?	Yes	
Ability to modify or replace user login and logout modules?	Yes	
Ability to check and enforce user privileges in customer operator interfaces?	Yes	
Ability to access user privileges in step preconditions and in test code modules?	Yes	
Source Control		
Does NI TestStand support source code control (SCC)?	Yes. NI TestStand implements source code control through the Microsoft SCC API. Files can be checked in and out from the Workspace and Project window.	
Can sequences be differentiated for code tracking?	Yes. NI TestStand has an advanced graphical differencing tool that shows all differences between any two sequences. In addition, the XML sequence file format can be differentiated with common diffing tools.	
Backwards Compatibility		
Current version is backward compatible with previous versions of the software?	Yes.	
List supported migration paths.	Sequences and tests from the LabVIEW and LabWindows/CVI Test Executive Toolkits can be migrated to NI TestStand. NI TestStand provide sequence converters that upgrade the sequence files from these previous packages to NI TestStand sequence files. Third parties can easily plug in additional converters to support migration from other legacy environments.	

Reporting/Result Management

Reporting		
ASCII Text Reports?	Yes	
HTML Reports?	Yes	
XML Reports?	Yes	
ATML Reports?	Yes	
Can the report content be configured?	Yes	
Can the location and naming of report files be configured?	Yes	
Can report generation be customized or replaced for the test station as a whole?	Yes	
Can report generation be customized or replaced on a per sequence file basis?	Yes	
Can test code modules add additional information to the test report?	Yes	
Can reports contain graphics and hyperlinks to diagnostic or repair procedures?	Yes. You must use the HTML or XML report formats.	
Can reports be generated on-the-fly?	Yes	
Result Collection		
Are results automatically collected for each step?	NI TestStand automatically collects the results for each step, including user-defined steps that contain extended result information	
Are results accessible from sequences, code modules, report generators, database loggers, and other tools?	NI TestStand results can be accessed in a sequence or from components written in any language.	
Can result collection be disabled?	Result collection can be disabled at the step, sequence, or test station level.	
Can code modules add additional result information?	A code module can add any amount of additional information to the results for a step. This information is accessible in report generators and other modules.	
Database Connectivity		
Can test results be automatically stored in database tables?	Yes	
Can data be set and retrieved from a database during execution outside of the results logging?	Yes. NI TestStand provides a complete set of database step types.	
Which databases are supported?	All ADO and ODBC compliant databases are supported. The default schema that the NI TestStand process model uses to store test results is compatible with Oracle, Access, SQL Server,	

	MySQL and Sybase.	
	Customization Features	
Operator Interfaces		
Can the operator interface program be customized or replaced?	Yes. NI TestStand includes operator interfaces written in LabVIEW, LabWindows/CVI, Visual Studio C++ (MFC for .NET), Visual Basic .NET, and C#. Each operator interface comes with full source code. NI TestStand also provides a complete set of user interface controls that reduce custom operator interface development by 90%.	
Which languages and environments can you use to develop operator interfaces?	You can use any language or environment that can call the NI TestStand ActiveX API. This includes LabVIEW, CVI, Visual C++, Visual Basic, and Visual J++.	
Can different operator interfaces execute the same test sequences?	Operator interface programs call the TestStand API to execute sequences. Thus, any operator interface program can execute any sequence file.	
Can users select the viewer application that the operator interface programs use to display report and limit files?	Yes	
Test Executive Engine API		
Does the test executive engine export a user-callable API?	NI TestStand is an open system that exports an ActiveX API that you can call from test code modules, operator interface programs, and third-party components. The TestStand API is similar in structure to the APIs exported by applications such as Word and Excel.	
What languages can call the test executive API?	Any language that can call an ActiveX server. This includes LabVIEW, CVI, C++, J++, and Visual Basic. NI TestStand includes C and C++ header and source files for calling the TestStand API from CVI and Visual C++	
Does the Test Executive API come with online help?	Yes	
What functionality does the test executive API export?	● All the functionality required to build operator interface programs such as the initiation, control and monitoring of sequence executions. ● All of the functionality required to enable custom tools to load, save, create, inspect, and edit sequence files. ● All of the functionality required to	

	enable test code modules to get or set the values of variables and properties, to detect termination requests during lengthy operations, to report status messages or progress indications to the operator interface, and to inspect or alter the state of execution.	
Definition of the Testing Process		
Does the test executive hard code a model of the testing process?	No. TestStand defines the testing process model as the sequence of actions that surround the execution of a test sequence. This includes operations such as identifying the UUT, displaying pass or fail notifications, generating test reports, or logging test information to a database. In TestStand, the occurrence and order of such actions in the testing process is defined by a sequence file that is designated as the station process model file. You are free to modify or replace this file.	
How many standard TestStand models are supplied?	Three. Sequence Model, Batch Model, and Parallel Model.	
Can the actions that the test executive automatically performs when it runs a test sequence be modified or replaced?	Yes. The process model defines actions such as identifying the UUT or generating a report. You can add to, change, or replace the actions that define your testing process by editing the process model sequence file. In addition, an action in the process model file that is marked as a callback can also be redefined in each individual test sequence file. For example, a sequence file can change or replace how its report is generated without altering the reports for other sequence files.	
Custom Sequence Editor		
Can the developer customize the Sequence Editor?	Yes. You can restrict development and execution capabilities using the built-in user management features. You can also customize the menu items as well as add your own custom tools to the Tools menu. In addition, TestStand includes example Custom Sequence Editors that you can customize written in LabVIEW, LabWindows/CVI, Visual Studio C++ (MFC for .NET), Visual	

	Basic .NET, and C#. Each custom sequence editor comes with full source code. TestStand also provides a complete set of user interface controls that reduce custom operator interface development by 90%.	
Custom Step/Test Types		
Can developers create reusable types of test steps that appear as if they are built into the test executive?	Yes	
Can custom step types define their own data, results, and run-time behavior?	Yes. You can create steps that accept custom inputs, return multiple results, and perform special processing. You can also create the editing dialog with which a user configures instances of your step type.	
In which programming languages can you develop custom step types?	Any language or environment that TestStand supports for test code module development.	
Can custom step types define source code templates to aid users in creating test code that is called by the step?	Yes, templates can be created for each code format NI TestStand supports.	
Localization Support		
Are user viewable text strings stored in separate files for each language?	Yes. TestStand stores all strings used by the sequence editor and operator interfaces is a set of ASCII files. You can add new languages files without recompiling or reconfiguring any TestStand components.	
Can string files be viewed and edited in a standard text editor?	Yes. All string files are in a standard .ini format.	
Can third-party components define new strings and redefine existing strings?	Yes. Third parties can supply their own string files so that their components can display localized strings.	
Are multi-byte character sets supported?	Yes	
Is Japanese Windows supported?	Yes	
Are product manuals available in multiple languages?	Some manuals are also available in Japanese.	

Documentation

Documentation and Online Help		
List the manuals and online help files.	● TestStand Quick Start Guide ● TestStand System and Architecture Overview Card ● Using TestStand	

	● Using LabVIEW with TestStand ● Using LabWindows/CVI with TestStand ● TestStand Reference Manual ● TestStand Online Help ● TestStand Course Manual (printed, available separately) ● TestStand Application Notes (ni.com/zone) ● Online Example Programs (ni.com/zone)	
Documentation Generation		
Can test sequence documentation be generated automatically?	Yes	
What documentation formats are available?	Text and HTML.	
Can the documentation format be modified?	Yes. The documentation generator is a component that calls the TestStand API to access the contents of sequence files. The source code is provided. You can modify or replace the documentation tool.	

Support and Training

Support		
Phone support available?	24 hr online technical support (ni.com/support) Live TestStand experts available 7am – 7pm (M-F)	
Technical discussion forum available?	National Instruments maintains a free TestStand technical discussion forum monitored by product developers. Join the forum at *http://forums.ni.com*	
Online knowledge base and file library on the web?	Yes, visit *www.ni.com/support/teststandsupp* and on the Zone at *zone.ni.com*	
Who is responsible for documentation, testing and long term software maintenance?	National Instruments maintains and supports all documentation, and releases new versions when necessary. Our large user base results in more complete testing of product.	
Training		
Training Course Available?	Yes. There are two training courses on TestStand taught regularly in major cities around the world. Courses in local languages are also available. Information on the TestStand training course is available online at	

	www.ni.com/training	
Course Available Onsite?	Yes. Onsite TestStand courses can be booked anywhere in the world.	
Course materials available separately?	Yes	
Is extended training available?	Yes. NI Certified TestStand Architects (CTAs) can work with you to define your system architectures and train you on implementation issues.	
An Open Industry Standard Architecture		
Third Party Support		
Can third party or in house developed tools be integrated into the test executive?	TestStand is a platform for interchangeable transportable test software components. Third parties can build add-on tools and integrate TestStand into full test systems.	
Can integrate with requirement management software like Telelogic DOORS or IBM Rational RequisitePro?	Yes, by using NI Requirements Gateway, NI TestStand can trace requirements coverage information from Telelogic DOORS, IBM Rational RequisitePro, MS Word, MS Excel, Adobe Acrobat and text files.	
Hardware Support		
Supports industry standard instrument drivers?	Yes, provides support to instrument and hardware I/O drivers through IVI steps, direct DLL calls, or through test modules written in any of the supported code formats	
Supports GPIB, RS-232, VXI, PXI, and plug-in instruments?	Yes, through test modules. LabVIEW, LabWindows/CVI, and Measurement Studio have built-in support for all major instrumentation hardware	
Switching support?	Yes, TestStand offers built-in switching integration within each test step. Automate your switching using TestStand and NI Switch Executive without writing a single line of code.	
Vendor Expertise and Experience		
Company experience in market?	National Instruments has been a part of the test and measurement market since 1976 and has supplied the test executive software market since 1993.	
Trained Integrators?	National Instruments Alliance Program includes expert TestStand integrators throughout the United States and the world.	
Purchasing Options		

Packaging		
Available configurations	● TestStand Development System. Comprehensive, customizable test management software. Includes customizable operator interfaces and sequence development environment. ● TestStand Base Deployment Engine. Includes TestStand execution engine and support files for deploying your system on a target computer. ● TestStand Debug Deployment Environment. Includes TestStand Deployment Engine and extended sequence debugging capabilities. Also includes a license to use LabVIEW or LabWindows/CVI on the target computer for debugging purposes only. ● Developer Suite, Automated Test Comprehensive suite of tools for building a test application, including: • TestStand Development System • LabVIEW - Graphical Programming • LabWindows/CVI – C for Virtual Instrumentation • Measurement Studio – Native tools for use with Visual Studio .NET • IVI Instrument Driver Library	