

TestStand Certification Overview

The National Instruments TestStand Certification Program consists of the following two certification levels:

- Certified TestStand Developer (CTD)
- Certified TestStand Architect (CTA)

The Certified TestStand Developer (CTD) certification represents core knowledge and skill in the design, development, and integration of production tests in the TestStand environment. A CTD demonstrates the following competencies in TestStand:

- Apply concepts, definitions, and configuration options
- Utilize TestStand data storage and data passing mechanisms
- Develop and debug tests using standard development environments

A CTD is a test developer who uses the TestStand framework to develop, analyze, debug, and deploy tests. The CTD certification is for developers who possess technical expertise in a certain product and wish to distinguish their TestStand knowledge, skills, and experience in test development.

CTDs may advance their credentials by gaining knowledge and experience in developing and customizing TestStand frameworks and distinguish themselves by attaining the CTA certification.

The Certified TestStand Architect (CTA) certification represents a professional skill level in the design, development, and deployment of a customized TestStand-based test system (framework) to meet a set of requirements. In addition to the competencies of the CTD, the CTA demonstrates the following:

- Skills in translating TestStand specifications to design documents
- Competency in the development and customization of the TestStand framework
- Experience in the design of a scalable, maintainable, and well-documented TestStand framework that utilizes established development guidelines and practices

A CTA is a TestStand framework developer who creates or customizes the TestStand framework to meet a set of test system requirements. The CTA certification is for developers who wish to distinguish their software development skills and technical experience in the TestStand system.

Candidates who want to certify at the CTA level must possess a valid CTD certification.



Note The CTD certification is a prerequisite to taking the CTA exam. There are no exceptions to this requirement.

TestStand Exam Overview

Product: TestStand Development System for Windows. Refer to ni.com/teststand for details on the features available in the TestStand Development System.

Exam Duration: 1 hour

Number of Questions: 40

Style of Questions: Multiple-choice

Passing grade: 75%

TestStand Version: 4.0



Note Every attempt is made to create exams that are independent of the software version. However, two to four test items may base the question or answer choices on the latest version of the software. NI recommends that you familiarize yourself with the latest version of the software before the exam. You can download the latest version of TestStand for evaluation from: ni.com/teststand.

The use of TestStand or any other external resources is prohibited during the exam. For assistance and wherever appropriate, screenshots are provided from the *NI TestStand Help*.

To maintain the integrity of the exam, you may not copy or reproduce any section of the exam. Failure to comply will result in failure.

In areas where the exam is deployed as a paper-based exam, detaching the binding staple will result in failure without evaluation.

Exam Logistics

United States and Europe: Both the TestStand certification exams can be taken at Pearson Vue test centers. The exams are computer-based and results are available immediately upon completion of the exam. Refer to www.pearsonvue.com for more details and scheduling.

Asia: The exam is paper-based, for which the evaluations and results take about 4 weeks. Please contact your National Instruments local office for details and scheduling.

For general questions or comments, email: certification@ni.com.

TestStand Developer Exam Topics

1. Terminology, concepts, and architectures
2. Creating, analyzing, and debugging tests
3. Data structures and data management
4. Overriding callbacks
5. Configuring TestStand
6. Multithreading
7. User management
8. Deployment
9. Best practices

TestStand Developer Exam Topics

Topic	Sub Topic
1. Terminology, concepts, and architecture	<ul style="list-style-type: none">a. Test terminology, system, and architecturesb. TestStand terminology, concepts, and componentsc. TestStand design and execution architecture
2. Creating, analyzing, and debugging tests	<ul style="list-style-type: none">a. Test creation:<ul style="list-style-type: none">i. Sequence filesii. Steps – types, settings, properties, and optionsiii. Code modulesiv. Subsequencesv. Sequence and sequence file propertiesb. Debugging and diagnostics<ul style="list-style-type: none">i. GUI componentsii. Toolsiii. Techniquesc. Information tools – Find and Diff
3. Data structures and data management	<ul style="list-style-type: none">a. Properties and containers<ul style="list-style-type: none">i. Static, dynamicii. Important propertiesiii. Accessing properties and subpropertiesb. Expressions<ul style="list-style-type: none">i. Syntax, types, and usageii. Accessing step propertiesc. Custom data types<ul style="list-style-type: none">i. Type palettes and locationsii. Handling type version conflictsd. Variables – local variables, parameters, file global variables, and station global variablese. Tools and techniques for importing and exporting properties
4. Overriding callbacks	<ul style="list-style-type: none">a. Callbacks in the execution modelb. Creating callback sequencesc. Process model callbacksd. Engine callbackse. Commonly used callbacks
5. Configuring TestStand	<ul style="list-style-type: none">a. Station optionsb. Adapter configurationsc. Search optionsd. Report optionse. Database options
6. Multithreading	<ul style="list-style-type: none">a. Multithreaded terminology and execution modelb. Multithreaded process models and synchronizationc. Multithreading caveatsd. Synchronizing step types

7. User management	<ul style="list-style-type: none"> a. User management concepts and tools b. User groups c. Privileges d. Synchronizing with external user lists and the Windows login
8. Deployment	<ul style="list-style-type: none"> a. Deployment requirements b. TestStand components needed for test system deployment c. Deployment process <ul style="list-style-type: none"> i. Workspace ii. Deployment utility
9. Best practices for test development	<ul style="list-style-type: none"> a. Designing modular test systems b. Creating and defining appropriate data structures and their scope c. Applying appropriate step types d. Using step groups e. Documenting the test system

TestStand Developer Exam Topic Details

1. Terminology, concepts and architecture

- a. Test terminology, system, and architectures
 - 1. Identify and differentiate between verification and validation tests.
 - 2. Define test system terminology and testing roles.
 - 3. Identify and define test system components.
 - 4. Identify and differentiate between test system setups.
- b. TestStand terminology, concepts, and components
 - 1. Identify and define the role and application of TestStand in test management.
 - 2. Identify the role, purpose, and application of TestStand components.
 - 3. Describe the interaction and communication of the TestStand components in the overall TestStand architecture.
 - 4. Use the environment to edit, debug, and execute test routines.
 - 5. Navigate the menus and set options of the components and dialog boxes to configure test settings.
- c. TestStand design and execution architecture
 - 1. Identify the components, functions, and communication mechanisms in the TestStand execution architecture.

2. Creating, analyzing and debugging tests

- a. Creating tests
 - 1. Identify and use the TestStand components to create test routine(s) to meet requirements.
 - 2. Open and create sequence files and sequences and set appropriate properties to meet test requirements.
 - 3. Distinguish among and apply the available step types.
 - 4. Add steps, code modules, and subsequences and configure their properties and settings to meet test requirements.
 - 5. Given a test situation and a specific component, determine the most appropriate settings for the key properties.
 - 6. Create and use step templates.
 - 7. Create, modify, and use code modules and interface through module adapters and parameters.
 - 8. Determine the best method to locate and configure paths to code modules.
- b. Debugging and diagnostics
 - 1. Identify and use TestStand windows and tools to debug and diagnose tests.
 - 2. Use breakpoints and the Watch View pane to view data items during execution.
 - 3. Given a problem scenario, determine the best method or technique to debug or diagnose the problem.
 - 4. Use the Find and Diff tools to analyze test routines.

3. Data structures and data management

- a. Properties and containers
 - 1. Identify and distinguish between static and dynamic properties.

2. Select and create or use properties to store test data.
3. Identify the best method to locate and retrieve test data by finding and accessing properties and subproperties.
- b. Expressions
 1. Identify key properties and containers and use them in expressions to evaluate and determine test progression.
 2. Determine the best way to access step data—step index, step ID, or step name.
- c. Custom data types
 1. Design and appropriately locate custom data types and containers.
 2. Handle type conflicts and versioning by using the appropriate dialog box options.
- d. Variables:
 1. Distinguish among the scope and default values of local variables, parameters, sequence file global variables, and station global variables.
 2. Select, create, and use the appropriate variable type to store test data.
- e. Tools and techniques for importing and exporting properties
 1. Apply and use the Import/Export Properties dialog box to store and retrieve variable and property data.
 2. Apply and use the Property Loader step type to dynamically update variable and property data.

4. Overriding callbacks

- a. Callbacks in the TestStand execution model
 1. Define callbacks and explain their purpose.
 2. Identify the components in the TestStand execution model that you can affect by overriding callbacks.
- b. Creating callback sequences
 1. Discuss the method for creating callbacks.
 2. Identify Process Model callbacks and determine the execution flow when a callback is overridden.
 3. Identify Engine callbacks and their applications.
 4. Identify the differences between a Process Model callback and an Engine callback.
 5. Given a test scenario, determine if you must override a callback and which callback must be overridden.

5. Configuring TestStand

- a. Determine the impact of and select options and settings in configuration dialog boxes and tabs for the following:
 1. Station options
 2. Adapter configurations
 3. Search options
 4. Report options
- b. Identify the purpose of and configure simple database logging using the options in the Database Options dialog box.

6. Multithreading

- a. Multithreaded terminology and execution model
 1. Determine the test station and multithreading requirements from the test requirements.
 2. Explain the multithreaded execution model in TestStand.
- b. Multithreaded process models
 1. Differentiate between the Parallel and Batch process models in terms of execution, timing, and synchronization.
 2. Select the most appropriate Synchronization step types to apply when creating a multi-UUT test.
- c. Multi-threading caveats
 1. Explain and differentiate among the following potential drawbacks that could happen in a multithreaded test system:
 - a) Race conditions
 - b) Resource conflicts
 - c) Deadlock
 2. Given a test scenario, determine the possible synchronization problem and apply the most appropriate Synchronization step type to resolve the problem.

7. User Management

- a. Given a set of requirements, determine the need for user management .
- b. Identify the tools and windows that add, delete, and configure users, groups, and their privileges.
- c. Identify the built-in user groups and the privileges associated with the user groups.
- d. Given a set of requirements, configure users and associate users with appropriate groups and privileges.
- e. For a user belonging to multiple groups, determine the user's privileges when the group's privileges change.
- f. Synchronize the TestStand user system with external data or the Windows user management system.

8. Deployment

- a. Determine the TestStand deployment requirements based on the target computer.
- b. Select an appropriate file naming and file path strategy to avoid cross-linking and enable TestStand to locate the files on the target computer?
- c. Determine the TestStand components required for the test system to execute on the target computer.
- d. Use the workspace file to organize the required components for deployment.
- e. Select and set appropriate options in the deployment utility to build a distribution image or installer.

9. Best practices for test development

- a. Design modular test routines that are cohesive, maintainable, and deterministic in a multithreaded test system.

- b. Design modular data structures (containers) to clearly define relationships among modules.
- c. Create subsequences and organize them in sequence files to create the modular test system design.
- d. Determine and limit the scope of data items to prevent cross-linking and race conditions.
- e. Use Setup and Cleanup step groups to allocate and de-allocate resources.
- f. Use Flow Control and Synchronization steps to control the test system and make the test execution deterministic.
- g. Document the test system at sequence file, sequence, and step levels.