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.

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 the 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 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 Architect Exam Topics

1. Test frameworks
2. TestStand API
3. Customizing steps
4. Customizing process models
5. Customizing user interfaces
6. Customizing database interaction
7. Design considerations

Note The Certified TestStand Architect exam is cumulative and includes topics from the Certified TestStand Developer exam.

TestStand Architect Exam Topics

Topic	Sub Topic
1. Test frameworks	a. Purpose of the test framework b. Components of the test framework c. Requirements for customizing the test framework
2. TestStand API	a. Purpose, functionality, and uses b. API organization—Objects, classes, and references c. Inheritance and containment d. PropertyObject class e. Using the TestStand API
3. Customizing steps	a. Custom step types b. Step templates
4. Customizing process models	a. Sequence types in the process model structure b. Selecting components for customization c. Common customizations to process models
5. Customizing user interfaces	a. TestStand user interfaces b. User interface controls c. User interface messages d. Front-End callbacks
6. Customizing database interaction	a. Options and procedure b. Modifying schemas c. Using the Logging property d. Database step types e. Customizing applications
7. Design considerations	a. Designing scalable and maintainable systems b. Locating custom functions and data c. Detecting, handling, and reporting errors d. Deploying a framework

TestStand Architect Exam Topic Details

1. **Test Frameworks**
   a. Purpose of the test framework
      1. Identify the usefulness and purpose of a test framework.
      2. Describe the purpose of the TestStand framework for various levels and roles in the testing process.

   b. Components of the test framework
      1. Describe the function of the following TestStand framework components
         a) Process model
         b) User interfaces
         c) Utility tools—data and database management tools
         d) Test building blocks—custom steps, data types, and templates

   c. Requirements for customizing the test framework
      1. Given a specification, create requirements for customizing the test framework.
      2. Identify the test framework components that are affected by the requirements and need customization.

2. **TestStand API**
   a. Purpose, functionality, and uses
      1. Describe the purpose of the TestStand API in customizing the test framework.
      2. Identify the TestStand components that can be affected by the use of the TestStand API.
      3. Identify the components from which you can use the TestStand API and the differences in the calling mechanisms.

   b. API organization
      1. Distinguish between TestStand API classes and objects.
      2. Identify the parent class of common TestStand objects.
      3. Apply and demonstrate the use of common properties and methods.
      4. Identify the class whose object is directly creatable and the methods available to create objects of other classes.
      5. Describe the ways you can use references to access TestStand objects.

   c. Inheritance and containment
      1. Describe the inheritance model in TestStand and use the inheritance model to access properties and methods of objects.
      2. Identify the difference between containment and inheritance.
      3. Identify the ways you can obtain references to objects.

   d. PropertyObject class
      1. Describe the significance of the PropertyObject class in the TestStand class hierarchy.
      2. Identify the TestStand objects that are directly represented as PropertyObjects.
      3. Use the PropertyObject to access dynamic and custom properties.

e. Using the TestStand API

1. Utilize Lookup strings to specify the access to properties, methods, and variables.
2. Utilize the Engine object reference to acquire references to other TestStand API objects.
3. Identify the differences in calling the TestStand API using the ActiveX adapter versus external code modules.
4. Discuss the pros and cons of and demonstrate using the TestStand API from expressions.

3. Customizing steps

a. Given a requirement, determine whether a custom step type or a step template is appropriate to implement the requirement.

b. Custom step types

1. Evaluate requirements and justify the creation of a custom step for a given application.
2. Use a custom step type to create a new step and customize its behavior.
3. Demonstrate the procedure to create a custom step type and determine its storage location.
4. Resolve type conflicts and versioning issues.
5. Configure custom steps by adding properties, modifying settings, and adding substeps.
6. Create a code module template for a custom step type.
7. Create a custom step type by combining two step types.

c. Step templates

1. Evaluate a requirement and justify the creation of a step template.
2. Demonstrate the procedure to create a step template.

4. Customizing process models

1. Given a set of test framework customization requirements, identify the need for customizing the process model.
2. Identify the different sequence types in a process model and their uses.
3. Given a set of test framework customization requirements, identify the process model components that need customization.
4. Customize a process model to display prompts to guide test operators during a test.
5. Use the TestStand API to customize data collection for results and report generation in the process model.

5. Customizing user interfaces

a. Differentiate between the simple and full featured user interfaces.

b. User interface controls

1. Use the ActiveX user interface controls to develop and customize user interfaces in LabVIEW or LabWindows/CVI.
2. Differentiate between using the TestStand API and the Active X user interface controls.
3. Identify the different types of available user interface controls and the functionality provided by them.

a) Manager controls

        b) Visible and invisible user interface controls

    4. Given a user interface requirement, determine the controls required to develop the user interface.

  c. Use predefined user interface messages or create user interface messages to interact with the TestStand process model.

  d. Create and use Front-End callbacks by invoking sequences from the user interface application.

6. **Customizing database interaction**

  a. Use the Database Options dialog box to configure the logging of TestStand data.

  b. Use key database concepts and terminology to identify TestStand database components for customization.

  c. Identify and use statements to modify schemas and manipulate and query data from database tables.

  d. Identify and use Logging container properties in expressions.

  e. Use database step(s) to execute database commands.

  f. Given a set of data logging requirements, select the most appropriate database structure and TestStand database customization tools to implement the requirement.

7. **Design considerations**

  a. Given a set of framework customization requirements, select the most scalable and maintainable TestStand design.

  b. Determine the most appropriate location to implement custom functions and store data.

  c. Limit the client sequence file to code specific to a particular type of UUT.

  d. Use process models and callbacks to implement core customizations.

  e. Use custom step types to perform a custom operation a step level.

  f. Abstract the user interface from the test framework.

  g. Categorize data and results.

  h. Define an error handling strategy to detect, handle, and report errors.

  i. Deploy the framework using TestStand deployment tools.