

Objectives and Tasks for:
Certified TestStand Architect

Certification ID	Certification Title	Job Description:
CTA	Certified TestStand Architect	Given an industrial testing need, the Certified TestStand Architect will provide efficient, comprehensive testing solutions to satisfy complex requirements

Task ID	Task	Objective ID	Objective
CTA-ASD-01	Exploit the modularity of the TestStand architecture in designing a TestStand system		
		CTA-ASD-01-01	Describe the role of a process model and callbacks in the TestStand architecture
		CTA-ASD-01-02	Describe the information available from the TestStand Programmer Help for: <ul style="list-style-type: none"> ▪ Structure of the API ▪ Information on TestStand properties and methods
		CTA-ASD-01-03	Given a set of requirements and deliverables, design and implement appropriate Sequence Files and Code Modules

Objectives and Tasks for:
Certified TestStand Architect

		CTA-ASD-01-04	<p>Given a set of requirements for a TestStand project, identify the role and customization necessary for the following TestStand Components to generate the optimum solution:</p> <ul style="list-style-type: none"> ▪ TestStand Engine ▪ User Interface <ul style="list-style-type: none"> ▪ Sequence Editor ▪ Operator Interface ▪ Process Model <ul style="list-style-type: none"> ▪ Process Model Callbacks ▪ Sequence Files ▪ Module Adapters ▪ Other Callbacks <ul style="list-style-type: none"> ▪ Engine Callbacks ▪ Front-End Callbacks ▪ TestStand API
		CTA-ASD-01-05	Given a set of conditions and required results, use the TestStand Help System to obtain the necessary information
		CTA-ASD-01-06	Given an object of a particular class, use it to obtain an object from another class
		CTA-ASD-01-07	Given a set of conditions and requirements, use the sequence context object to obtain the requested information
		CTA-ASD-01-08	Describe the Lookup String and its association with properties
		CTA-ASD-01-09	Given a set of requirements, create an appropriate Lookup String to access specific data

Objectives and Tasks for:
Certified TestStand Architect

		CTA-ASD-01-10	Describe the process by which the ActiveX Automation Adapter can be used to make TestStand API calls from sequences
		CTA-ASD-01-11	Use the ActiveX Automation Adapter from within a specific sequence for calling the TestStand API to satisfy a given set of requirements
CTA-ASD-02	Utilize the design features of TestStand Process Models to create optimum test solutions		
		CTA-ASD-02-01	Define the role of the Process Model in eliminating redundancy from testing operations
		CTA-ASD-02-02	Describe the advantages of using the ability of a model callback to override components of the process model
		CTA-ASD-02-03	Describe the method and advantages for making a process model respond to configuration parameters
		CTA-ASD-02-04	Describe the advantage of using shipping TestStand process models as starting points
		CTA-ASD-02-05	Given a set of conditions and requirements, create a TestStand solution that minimizes redundancy of standard testing operations
		CTA-ASD-02-06	Given a set of conditions and requirements, select the appropriate process model to be customized

Objectives and Tasks for:
Certified TestStand Architect

		CTA-ASD-02-07	Given a set of conditions and requirements for a TestStand project, implement a solution that incorporates the optimum combination of existing and customized components necessary to generate the optimum solution
CTA-ASD-03	Incorporate Callbacks in a TestStand implementation to perform actions under specific circumstances		
		CTA-ASD-03-01	Analyze a set of conditions to determine the need for a model callback, and if needed, whether a predefined or new callback is necessary
		CTA-ASD-03-02	Describe the flexibility, modularity, and control that callbacks provide in the design of a TestStand system
		CTA-ASD-03-03	Given a set of requirements and deliverables, design and implement appropriate callbacks in a TestStand project
		CTA-ASD-03-04	Generate a Model Callback to customize the test process for a specified test sequence
		CTA-ASD-03-05	Create an Engine Callback in a sequence to modify the behavior of that sequence to satisfy a given need
		CTA-ASD-03-06	Create an Engine callback to satisfy a given set of requirements

Objectives and Tasks for:
Certified TestStand Architect

		CTA-ASD-03-07	Explain the proper conditions under which to use a Front End Callback
		CTA-ASD-03-08	Given a set of conditions and requirements, incorporate Front End Callbacks in a TestStand solution
CTA-ASD-04	Create Custom Types for TestStand implementations		
		CTA-ASD-04-01	State the design considerations involved in creating custom types for TestStand
		CTA-ASD-04-02	Describe encapsulation as it applies to creating custom types in TestStand
		CTA-ASD-04-03	Given a customized TestStand type and its associated TestStand application, analyze the appropriateness of the type for that application
		CTA-ASD-04-04	Given a set of requirements and deliverables, design and implement appropriate Step Types
		CTA-ASD-04-05	Given a set of data requirements for an application, create an appropriate custom data type
CTA-ASD-05	Design and Implement TestStand Operator Interfaces.		

Objectives and Tasks for:
Certified TestStand Architect

		CTA-ASD-05-01	Describe the design considerations for an Operator Interface
		CTA-ASD-05-02	Given a set of specifications, identify the appropriate Operator Interface to be modified and justify your decision <ul style="list-style-type: none"> ▪ Simple Operator Interfaces ▪ Full Operator Interfaces
		CTA-ASD-05-03	Given a set of requirements and deliverables, design and implement an appropriate Operator Interface for a TestStand implementation
		CTA-ASD-05-04	Identify and describe the modular components of a TestStand Operator Interface that are customizable
		CTA-ASD-05-05	Describe the internal architecture of the Operator Interfaces that ship with TestStand
		CTA-ASD-05-06	Given a set of conditions and requirements, implement a TestStand Operator Interface that communicates with the Process Model
		CTA-ASD-05-07	Describe the methods used for communications from: <ul style="list-style-type: none"> ▪ Process Model to the Operator Interface ▪ Operator Interface to the Process Model
		CTA-ASD-05-08	Given a set of requirements, modify a default TestStand Operator Interface to display information in a multithreaded system

Objectives and Tasks for:
Certified TestStand Architect

CTA-ASD-06	Implement Error Handling in TestStand Customizations		
		CTA-ASD-06-01	Describe the default TestStand runtime error handling mechanism
		CTA-ASD-06-02	Explain where runtime error information is stored for: <ul style="list-style-type: none"> ▪ Steps ▪ Sequences
		CTA-ASD-06-03	Differentiate between fatal and non-fatal errors and describe the resulting actions when they are encountered
		CTA-ASD-06-04	Compare and contrast Step.Result.Error and RunState.SequenceError
		CTA-ASD-06-05	Given a TestStand implementation, create examples which fully validate the error handling scheme
		CTA-ASD-06-06	Given a set of requirements and deliverables, design and implement appropriate error handling in a TestStand project
CTA-ASD-07	Design and implement a multithreaded application in TestStand		
		CTA-ASD-07-01	Explain the benefits of using a multithreaded test system
		CTA-ASD-07-02	Given a set of conditions and system requirements, determine whether a multithreaded TestStand solution is desired

Objectives and Tasks for:
Certified TestStand Architect

		CTA-ASD-07-03	Describe the Thread and Execution objects as they relate to TestStand
		CTA-ASD-07-04	Differentiate between synchronous and asynchronous test systems
		CTA-ASD-07-05	Explain the precautions to be taken when designing a multithreaded TestStand system to avoid multithreading problems including: <ul style="list-style-type: none"> ▪ Deadlock ▪ Priority Inversion ▪ Thread Starvation
		CTA-ASD-07-06	Create a synchronous test system that satisfies a given set of conditions and requirements
		CTA-ASD-07-07	Create an asynchronous test system that satisfies a given set of conditions and requirements
		CTA-ASD-07-08	Given a set of conditions and requirements, design a TestStand customization that appropriately implements multithreading capability
		CTA-ASD-07-09	Evaluate the susceptibility of a given TestStand implementation for multithreading problems <ul style="list-style-type: none"> ▪ Deadlock ▪ Priority Inversion ▪ Thread Starvation

Objectives and Tasks for:
Certified TestStand Architect

CTA-ASD-08	Incorporate good design practices into TestStand implementations		
		CTA-ASD-08-01	Given a set of conditions and requirements, create a requirements map to be used on the design solution
		CTA-ASD-08-02	Define the scope considerations within the Process Model design <ul style="list-style-type: none"> ▪ Client Sequence File use of Model Callbacks to modify the Process Model at Runtime ▪ Engine Callbacks affect all Client Sequence Files executed within a Process Model ▪ Specification of the desired Process Model by a Client Sequence File
		CTA-ASD-08-03	Define the scope considerations for Step design options within a TestStand application <ul style="list-style-type: none"> ▪ Engine Callbacks affect all steps of all types in Client Sequence ▪ Property Loader can affect all properties in a specific sequence ▪ Step Properties affect all steps of a specific step type
		CTA-ASD-08-04	Given a set of requirements, design a step type that minimizes external code usage and takes advantage of built-in TestStand capabilities

Objectives and Tasks for:
Certified TestStand Architect

		CTA-ASD-08-05	<p>Discuss common good programming practices for the following:</p> <ul style="list-style-type: none"> ▪ SequenceContext ▪ ActiveX Object Handles ▪ Free your resources ▪ Example Sequences <ul style="list-style-type: none"> ▪ Hardware Independent ▪ Mirror target application ▪ Test Custom features and/or Requirements ▪ Error Checking
		CTA-ASD-08-06	<p>Describe the effects of the following topics on a TestStand system</p> <ul style="list-style-type: none"> ▪ File Structure ▪ Relative vs. Absolute Paths ▪ Order of Step Actions ▪ Customizing Default TestStand Data Types ▪ Application Distribution ▪ Monitoring Execution Status
		CTA-ASD-08-07	<p>Given a set of design goals, use a modular approach in the design of a TestStand project</p>
		CTA-ASD-08-08	<p>Incorporate accepted good programming practices into a TestStand project, including:</p> <ul style="list-style-type: none"> ▪ Code reuse ▪ Appropriate use of the TestStand Executive ▪ Proper error handling ▪ Proper use and reuse of objects in the TestStand API ▪ Creation of example sequences for system testing ▪ Appropriate use of the Process Model

Objectives and Tasks for:
Certified TestStand Architect

CTA-ASD-09	Generate a Hardware Interface Layer (HIL) for a TestStand application		
		CTA-ASD-09-01	Differentiate between the Functional based and Instrument based HIL Designs
		CTA-ASD-09-02	Explain the benefits of incorporating an HIL within a TestStand system
		CTA-ASD-09-03	Describe the IVI Step Type as an example of an HIL
		CTA-ASD-09-04	Explain the function of the Instrument Session Manager
		CTA-ASD-09-05	Explain the impact of the following HIL design considerations <ul style="list-style-type: none"> ▪ Information required by various layers of drivers ▪ Bottom-up design methodology ▪ Appropriate function parameters
		CTA-ASD-09-06	Given a set of requirements, design a functional-based Hardware Interface Layer
		CTA-ASD-09-07	Given a set of requirements, design an instrument-based Hardware Interface Layer
		CTA-ASD-09-08	Given a set of conditions, incorporate the Instrument Session Manager into a TestStand implementation