



UFT120 – Unified Functional Testing 12.0 Essentials

Instructor-Led Training

For version 12.0

Overview

This course provides a comprehensive understanding of how to use the Unified Functional Testing (UFT) 12.0 application as an automated functional testing tool.

Beginning with a record and playback session, participants will learn how to create new scripts. They then explore enhancements, which add intelligence and flexibility to a script, including synchronization, checkpoints, and parameterization.

Later in the course, participants adopt principles frequently followed by developers as they design more sophisticated test scripts. By creating reusable actions, function libraries, and shared object repositories, and by learning to manipulate dynamic objects and data, participants will be armed with a skill set applicable for testing nearly any type of application.

Intended Audience

This course is intended for:

- New users of Unified Functional Testing (UFT), who need to automate manual testing and verification in a short amount of time. UFT 12.0 is the HP Software upgrade of Quick Test Professional 11.0.
- Quality assurance engineers who assume technical lead roles in the use of QuickTest Professional/UFT applications.

Duration: 5 Days

Software Version: 12.0

Prerequisites

Working knowledge of:

- Windows
- Web sites and browsers

Any experience with:

- Programming languages
- Scripting languages
- Writing macros

Course Objectives

At the end of the course, you should be able to:

- Create basic scripts from a manual test case
- Enhance basic tests with synchronization and verification
- Parameterize tests to run with multiple sets of data.
- Create and reuse modular actions
- Use the Object Repository
- Use debugging tools
- Write custom checkpoints to create more precise verification points within a test
- Use the Object Repository Manager
- Resolve object recognition problems
- Identify when the Editor is useful.
- Retrieve and use the properties of an object.
- Create programmatic descriptions.
- Create tests that include VBScript.
- Retrieve data from application objects.
- Use the Data Table object to store run-time data and drive actions.
- Create scripts that access data from external sources.
- Create new subroutines and functions.
- Use the Function Library editor.
- Identify when to handle exceptions programmatically.

Related Courses

- ALM120 – Application Lifecycle Management 12.0 Essentials
- UFT011T – Unified Functional Testing 12.0 Interactive Training Created by ART

Certification Exams

- HP AIS – Application Lifecycle Management
- HPASE – Application Lifecycle Management Implementations

UFT160 – Fundamentals of Unified Functional Testing

Module 1 – Course Overview

- Identify the contents and objectives of the course
- Define the class schedules and class logistics
- Identify the related courses
- Discuss the lab environment details

Module 2 – Software Overview

- Describe the advantages of UFT as a testing tool
- Preview the UFT Start Page and Help menus
- Recognize the sample applications used in the labs
- Identify resources for getting assistance

Module 3 – Preparing to Record

- Identify functional testing principles, and the benefits of automated testing
- Navigate the typical GUI testing workflow
- Document the steps of a business process
- Prioritize business processes using effective criteria
- Gather sufficient test data
- Prepare the test environment for automated testing

Module 4 – Creating a Basic Test

- Create a basic test from a manual test case
- Run a test and check for errors
- Save a test
- View test results

Module 5 – Working with Objects

- Identify objects
- Define a UFT for GUI Testing object
- Identify objects in UFT for GUI Testing
- Use the object repository to manage objects in UFT for GUI Testing

Module 6 – Adding Synchronization

- Define synchronization in UFT for GUI Testing
- Identify the uses of synchronization in UFT for GUI Testing
- Add a synchronization step for a specified object

Module 7 – Verifying with Standard Checkpoints

- Define standard checkpoints
- Add standard checkpoints to a test
- Use a regular expression to add flexibility to a standard checkpoint

Module 8 – Using Parameters

- Identify and use different parameter types
- Insert an input parameter
- Insert an output parameter
- Parameterize a checkpoint
- Evaluate test results for iterative tests
- Automatically parameterize steps

Module 9 – Building Multiple, Reusable Actions

- Identify actions in GUI testing
- Identify action types
- Identify action and test iterations
- Identify calls to existing actions and copies of actions
- Share values using the global data table
- Call actions with parameters
- Store action return values
- Create multiple actions from a single action
- Create a new action
- Call a reusable action from another test
- Use local and global data sheets
- Resolve missing actions

Module 10 – Adding Steps Without Recording

- List the types of steps that can be added to a test without using the record feature
- Use conditional statements in a test
- Use the Step Generator
- Use the reporter object to report events in the test results

Module 11 – Creating Tests on a Web Application

- Record and run a test on a web application
- Insert a text checkpoint in a test for a web application

Module 12 – Writing Custom Checkpoints

- Create a custom checkpoint
- Compare captured parameter values with expected values
- Use debug tools to investigate the causes of failure
- Verify that a reported error message reflects the state of the test

Module 13 – Utilizing a Shared Object Repository

- Identify the types of object repositories
- Manage shared object repositories using the Object Repository Manager
- Use visual relation identifiers

UFT220 – Working with Unified Functional Testing 12.0

Module 1 – Course Overview

- Identify the contents and objectives of the course
- Define the class schedules and class logistics
- Identify the related courses
- Discuss the lab environment details

Module 2 - Introduction to the Editor

- Identify the advantages of the Editor
- Translate steps between the Keyword view and the Editor view
- Explain how VBScript and objects are relevant to UFT for GUI Testing
- List common test objects and methods used in UFT for GUI Testing

Module 3 – Using the Editor

- Use the step generator to create new steps in your text
- Enter steps manually in the Editor
- Use statement completion (IntelliSense)
- Trace and debug tests

Module 4 – Using VBScript

- Create constants and variables to hold important values
- Format steps and create comments for clarity
- Use operators to modify or compare values
- Build conditional statements
- Call built-in functions from the VBScript function library

Module 5 – Working with Object Properties

- Retrieve any property of any object in the application during a test run
- Retrieve and set properties in the object repository
- Use additional properties of the Report utility object

Module 6 – Working with Dynamic Data

- Retrieve useful properties from list-type objects
- Describe the VBScript looping statements
- Use additional methods to explore the DataTable

Module 7 – Working with Dynamic Objects

- Identify types of dynamic objects
- Manage dynamic properties of objects in the object repository
- Build a programmatic description for an object
- Create and use a Description object
- Retrieve a collection of child objects from a parent object

Module 8 – Retrieving External Data

- Build scripts that access data from external sources
- Import data from and export data to a Microsoft Excel worksheet
- Use the Connection and RecordSet objects to query a database
- Import and export data to text files

Module 9 – Handling GUI Testing Exceptions

- Describe exception in a test
- Handle positive and negative test data
- Handle exceptions programmatically

Module 10 – Creating New Procedures

- Identify the advantages of creating a procedure in a test
- Create new subroutines and functions
- Build a function library and associate it with a test
- Register a procedure with an object class
- Use a function as a recovery operation in a Recovery Scenario