Course Data Sheet

# UFT120 – Unified Functional Testing 14.x Essentials

| | |
|---|---|
| Course No.: UFT120-140 | Category/Sub Category: Application Functional Testing/Unified Functional Testing-QTP |
| For software version(s): 14.0<br>Software version used in the labs: 14.0 | Course length: Five days |
| Delivery formats: Instructor Led (ILT) and Virtual Instructor Led (VILT) | Training is available as a private session onsite or as a public offering. |
| To order visit: Software Education | |
| For a preview of this course: UFT120 – Unified Functional Testing 14.x Essentials | |

## Course Description

This course provides a comprehensive understanding of how to use the Unified Functional Testing (UFT) version 14.0 application as an automated functional testing tool. Beginning with record and playback, participants learn how to create new automated tests. They then explore enhancements, including synchronization, checkpoints, parameterization, reusable actions, function libraries, and shared object repositories.

Included is an introduction to UFT for API testing, which contains an extensible framework for the construction and execution of functional tests of headless systems (systems that do not have a user interface).

## Audience/Job Roles

Quality Assurance engineers or any new users of UFT.

## Course Objectives

Upon successful completion of this course, you should be able to:

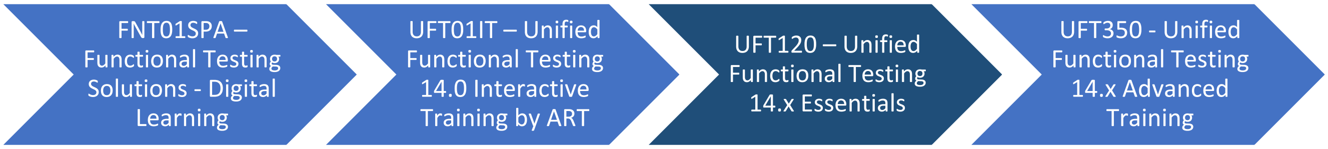
- Record and create scripts using Unified Functional Testing (UFT)
- Enhance the scripts with synchronization, parametrization, and checkpoints
- Create tests on Web applications
- Use UFT to automate GUI and API tests
- Test Web services with UFT API
- Add additional UFT features to make the test robust and provide better coverage

## Prerequisites/Recommended Skills

To be successful in this course, you should have the following prerequisites or knowledge:

- Working knowledge of Windows and web browsers
- Experience with programming or scripting languages

## Learning Path



FNT01SPA – Functional Testing Solutions - Digital Learning

UFT01IT – Unified Functional Testing 14.0 Interactive Training by ART

UFT120 – Unified Functional Testing 14.x Essentials

UFT350 - Unified Functional Testing 14.x Advanced Training

## Certification

- ASP – UFT-120-125 – Unified Functional Testing v12.5 (Available 1 November 2017)

## Course Topics

| Modules | Objectives |
|---|---|
| **Module 1: Course Overview** | ▪ Identify the contents and objectives of the course<br>▪ Define the class schedules and class logistics<br>▪ Identify the related courses<br>▪ Discuss the lab environment details<br>▪ Introduce SaaS environment. |
| **Module 2: Software Overview** | ▪ Describe the advantages of UFT as a testing tool<br>▪ Preview the UFT Start page and Help menus<br>▪ Recognize the sample applications used in the labs<br>▪ Identify resources for getting assistance |
| **Module 3: Preparing to Record** | ▪ Identify functional testing principles and the benefits of automated testing<br>▪ Navigate the typical GUI testing workflow<br>▪ Document the steps of a business process<br>▪ Prioritize business processes using effective criteria<br>▪ Gather sufficient test data<br>▪ Prepare the test environment for automated testing |
| **Module 4: Creating a Basic Test** | ▪ Create a basic test from a manual test case<br>▪ Run a test and check for errors<br>▪ Save a test<br>▪ View test results |
| **Module 5: Working with Objects** | ▪ Identify objects<br>▪ Define a UFT for GUI Testing object<br>▪ Identify objects in UFT for GUI Testing<br>▪ Use the Object Repository to manage objects in UFT for GUI Testing<br>▪ Resolve object identification issues |
| **Module 6: Utilizing a Shared Object Repository** | ▪ Identify the types of object repositories<br>▪ Manage shared object repositories using the Object Repository Manager<br>▪ Use visual relation identifiers |
| **Module 7: Adding Synchronization** | ▪ Define synchronization in UFT for GUI Testing<br>▪ Identify the uses of synchronization in UFT for GUI Testing<br>▪ Add a synchronization step for a specified object |
| **Module 8: Verifying with Standard Checkpoints** | ▪ Define standard checkpoints<br>▪ Add standard checkpoints to a test<br>▪ Use a regular expression to add flexibility to a standard checkpoint |
| **Module 9: Using Parameters** | ▪ Identify and use different parameter types<br>▪ Insert an input parameter<br>▪ Insert an output parameter<br>▪ Parameterize a checkpoint<br>▪ Evaluate test results for iterative tests |

| | |
|---|---|
| **Module 10: Building Multiple, Reusable Actions** | ▪ Identify actions in GUI testing<br>▪ Identify action types<br>▪ Identify action and test iterations<br>▪ Identify calls to existing actions and copies of actions<br>▪ Share values using the global data table<br>▪ Call actions with parameters<br>▪ Store action return values<br>▪ Create multiple actions from a single action<br>▪ Create a new action<br>▪ Call a reusable action from another test<br>▪ Use local and global data sheets<br>▪ Resolve missing actions |
| **Module 11: Adding Steps Without Recording** | ▪ List the types of steps that can be added to a test without using the record feature<br>▪ Use conditional statements in a test<br>▪ Use the Step Generator<br>▪ Use the reporter object to report events in the test results |
| **Module 12: Creating Tests on a Web Application** | ▪ Record and run a test on a web application<br>▪ Insert standard checkpoints on web objects<br>▪ Insert a text checkpoint in a test for a web application |
| **Module 13: Testing web services with UFT API** | ▪ Define Service Oriented Architecture (SOA)<br>▪ Identify components of SOA<br>▪ Define the terminology and principles surrounding component testing<br>▪ Navigate the UFT UI for API testing<br>▪ Create a basic Service Test (API Test) in UFT |
| **Module 14: Using UFT API** | ▪ Import a service<br>▪ Define test step inputs and outputs<br>▪ Identify data drive steps<br>▪ Work with controls flows<br>▪ Use the Results Viewer |
| **Module 15: Enhancing UFT API Testing** | ▪ Use checkpoints and reporting<br>▪ Configure database validation with UFT<br>▪ Configure security with UFT |

| | |
|---|---|
| **Appendix A: Using Database Checkpoints** | ▪ Identify the purpose of a database checkpoint<br>▪ Create a Structured Query Language (SQL) statement using Microsoft Query<br>▪ Create a database checkpoint<br>▪ Parameterize a database query |
| **Appendix B: Significant Checkpoints for GUI Testing** | ▪ Create table checkpoints<br>▪ Create test and text area checkpoints<br>▪ Create file content checkpoints<br>▪ Create XML checkpoints |
| **Appendix C: Object Identification Techniques** | ▪ Configure object identification<br>▪ Describe the mandatory and assistive properties<br>▪ Use ordinal identifiers<br>▪ Use smart identifiers<br>▪ Describe when to use Smart Identification<br>▪ Use the Smart Identification process<br>▪ Describe how UFT for GUI Testing uses Smart Identification – Use Case Scenario |
| **Appendix D: Using Recovery Scenarios** | ▪ Identify exceptions in a test<br>▪ Create a recovery scenario<br>▪ Associate a recovery scenario with a test<br>▪ Set an optional step in a test |

This page is intentionally left blank.