

# Unified Functional Testing 14.5x Advanced V(ILT) - (UFT ONE)

Learn via: **Classroom / Virtual Classroom / Online**

Duration: **2 Day**

**Overview** This course provides a comprehensive understanding of how to leverage and enhance the various features of Unified Functional Testing (UFT) by writing codes in the Editor. In this course, participants adopt principles frequently followed by developers as they design more sophisticated test scripts. By learning to manipulate dynamic objects and data, participants will be armed with a skill set applicable for testing nearly any type of application.

## **Prerequisites**

- Have working knowledge of Windows, websites, and browsers
- Understand the basics of UFT, programming languages, scripting languages, and macros

## **Who should attend?**

- Users who are familiar with the basics of UFT
- Quality assurance engineers who assume technical lead roles in the use of QuickTest Professional/UFT applications

## **What You Will Learn**

- List the main components of the Editor
- Identify when to use the Editor
- Retrieve and use the properties of an object
- Create programmatic descriptions
- Create tests that include VBScript
- Retrieve data from application objectsUse 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

## **Outline**

### **Module 1: Course Overview**

- Describe the Course objectives, course schedule, and class logistics
- Goals for the course
- Recognizing fellow participants
- Prerequisites
- Lab environment overview

### **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
- List common test objects and methods used in UFT

### **Module 3: Using the Editor**

- Use the step generator to create new steps in your test
- 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 Reporter 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
- Perform DataTable operations
- Use the Connection and RecordSet objects to query a database
- Import and export data to text files

### **Module 9: Handling GUI Testing Exceptions**

- Describe exceptions in a test
- Describe how to handle exceptions with the test settings
- Describe how to handle exceptions with Recovery Scenarios
- Handle exceptions programmatically with VBScript
- Handle positive and negative test data

### **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