

QA-QUICK TEST PRO(QTP)-ADVANCE

Introduction

Quick Test Professional (QTP) is an automated functional Graphical User Interface (GUI) testing tool that allows the automation of user actions on a web or client based and desktop computer application. It is primarily used for functional regression test automation. QTP uses a scripting language built on top of VBScript to specify the test procedure, and to manipulate the objects and controls of the application under test.

As part of a functional test suite, it works together with Mercury Interactive WinRunner and HP Quality Center and supports enterprise Quality Assurance.

QTP is UI automation software designed for testing Web-based and Windows based applications running on Microsoft Windows. Like other test automation tools, it works by identifying the objects in the application UI or a web page and performing the desired operations on them (like mouse clicks or keyboard events); it can also be used to capture object properties like name or handler ID etc. To perform these actions, QTP uses a scripting language built on top of VBScript to specify the test procedure, and to manipulate the objects and controls of the application under test. To perform sophisticated actions, users may need to manipulate the underlying VBScript.

Though QTP is usually used for “UI Based” Test Case Automation, it can automate some “Non-UI” based Test Cases. e.g. API (Certification testing), Database Testing, etc.

Course Contents

Identify the advantages of Expert View

Translate steps between Keyword View and Expert Views

Enter test steps in Expert View

Retrieve and use the properties of an application object

Use constants and variables in tests

Identify application objects with programmatic descriptions

Create tests that include VBScript operators

Functions and statements

Retrieve data from application objects

Describe and use various VBScript looping statements

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

Create and associate a library of functions

Use the Function Library editor

Identify when to handle exceptions programmatically

Lab Exercise(s)/ Practice sessions

IIBS