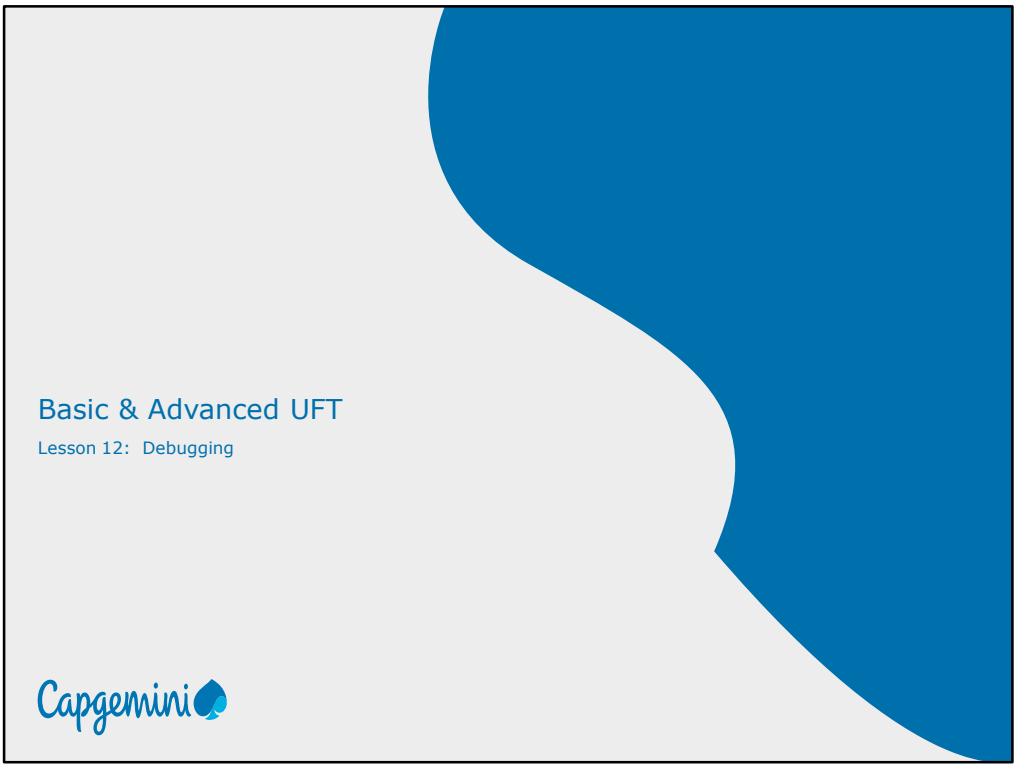
Basic & Advanced UFT
Lesson 12: Debugging
Capgemini

Lesson Objectives

- Script Debugging
- Debug Commands
- Debug Panes
- Debug Commands

Add instructor notes here.

Script Debugging

- Script debugging is a process, involving careful examination of the code line by line while executing the script with an objective to see the actions performed by the script at every step.
- This is required to fix a script which does not perform as expected.
- Debugging is a process of eliminating the bugs in UFT scripts.
- For starting the debugging process, go to "Run > Debug from Step" option.
- Other options for debugging can be viewed from "View > Debug". The options are Breakpoints, Call Stack, Loaded Modules, Threads, Local Variables, Console & Watch

Add instructor notes here.

Script Debugging (contd..)

There are three types of debugging processes:

1. Debugging by "Step Into": When we select "Run > Step Into" option, we can see if a function being executed is performing as expected.
   - This will open the function desired to be debugged in "Read Only" mode and we can keep on hitting the "F11" key on the keyboard to view the execution of every line of the function.

2. Debugging by "Step Over": This option is selected when we are sure that the function is performing as expected & we don't want to view the execution of the function.
   - We can hit "F10" key to execute the entire function without stopping and will stop for our next command at the beginning of the next line after the function call.

Add instructor notes here.

Script Debugging (contd..)

3. Debugging by "Step Out": This option is selected when we are in the function debug and we are sure that the function is performing as expected & we don't want to debug the execution of the entire function.
   - We can hit "Shift +F11" keys to execute the remaining statements in the function without stopping and will stop for our next command at the beginning of the next line after the function call.

Add instructor notes here.

Debug Commands

- **Run to Step (Ctrl + F10)**

  You can instruct UFT to run from the beginning of the test or action (Editor only)—or from the current location in the test or action—and to stop at a particular step. This is similar to adding a temporary breakpoint to a step. For example, if you want to begin debugging your test or action from a particular step, you may want to run your test or action to that step, as this opens your application to the relevant location

Add instructor notes here.

Debug Commands (contd..)

- **Debug from Step**

  You can instruct UFT to begin your debug session from a particular step instead of beginning the run at the start of the test or action. Before you start debugging from a specific step, make sure that the application is open to the location where you want to start debugging. You can begin debugging from a specific step in your test or action when editing a test or action

Add instructor notes here.

Debug Commands (contd..)

- **Run from Step (Ctrl + F5)**

  You can use the Run from Step option to run a selected part of your component from the selected step to the end of the component. This enables you to check a specific section of your application or to confirm that a certain part of your component runs smoothly

Add instructor notes here.

Debug Panes

The View > Debug options includes the following panes:

- **Breakpoints:** It provides information about breakpoints inserted into your GUI actions, scripted GUI components, function libraries or user code files and navigate directly to the breakpoint location in the relevant document.
- **Call Stack:** It enables you to view information about the methods and functions that are currently on the call stack of your test, component, function library, or user code file or the context in which the run session was paused.
- **Loaded Modules:** This debug pane enables you to view information about the .dll files loaded and executed as an API test runs.

Add instructor notes here.

Debug Panes (contd..)

- **Threads:** This pane enables you to view information about the threads currently running as part of the run session.
- **Local Variables:** This debug pane displays the current values and types of all variables in current the context of your document.
- **Console:** This debug pane enables you to run lines of VBScript code or C# code in your suspended run session.
- **Watch:** This debug pane enables you to view the current values and types of selected variables, properties, and VBScript or C# expressions in your suspended run session.

Add instructor notes here.

Debug Commands (can be viewed from Run tab in Menu bar)

- **Pause :** Pauses the Run/debug session
- **Add to Watch :** Adds the selected item to the Watch tab.
- **Insert/Remove Breakpoint :** Sets or clears a breakpoint in the test
- **Enable/Disable Breakpoint :** Enables or disables a breakpoint in the test
- **Clear All Breakpoints :** Deletes all breakpoints in the test
- **Enable/Disable All Breakpoints :** Enables or disables all breakpoints in the test

Summary

In this lesson you have learnt

- How debug the script
- What are the different options available to Debug the script

Add the notes here.