Quick Test Professional
Lab

Contents

1. CREATING A BASIC TEST

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Lesson 4: Creating a Basic Test

Lab Purpose

In this lab you will

- Record and execute a new script
- Add an initial condition to a script
- Save a script

Check off each task below as you complete it:

☐ **Exercise 1.1 - Record:**

- ☐ Start a new script
- ☐ Record the flow to create a new order

  Start the script from the "From" field.
- ☐ Execute your script at least two times.

  Did the script pass or fail. Why?

  _____

  What is required to fix the script?

  _____

☐ Exercise 1.2 – Initial Condition:

- ☐ Add an initial condition to your script

  What is the initial condition?

  _____

  Where is the initial condition located in your script?

  _____

  View both Expert and Keyword views.

- ☐ Execute your script at least 2 times

  Review the report

- ☐ Save your script as “Create Order”

You have now completed Lab 1, Creating a Basic Test.

You may now continue with the next lesson in the Quick Test Professional course.

2. OBJECT REPOSITORY (OR)

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 8: Object Repository

Lab Purpose

In this lab you will

- Open the Object Repository
- Change the logical name of an object
- Change the physical description of an object
- Add an object to the Object Repository

Check off each task below as you complete it:

☐ **Exercise 5.1 – Learn Objects by Recording, Change Logical Name:**

- ☐ Start a new script
- ☐ Record the flow of creating a new order

  Start the script with an Initial Condition

  Use Synchronization where necessary
- ☐ Open the Object Repository (OR)

  Into which OR did UFT learn the objects?

  _____

  Are you unhappy with any of the objects names?

  _____

  Which one?

  _____

  Why did UFT learn it like that?

  _____

Change the Logical Name of this object to a meaningful name. Where was the name changed in addition to the OR?

_____

- ☐ Save the script as “OR”

☐ Exercise 5.2 – Parameterized OR – Regular Expression:

- ☐ Add a verification that the "completed" message appeared

  Learn the object using its **text** property

- ☐ Execute the script

  Why did the script fail?

  _____

  Do you need to change the Logical Name or the Physical Description of the "completed" object in the OR?

  _____

- ☐ Use a Regular Expression to fix the problem in the OR
- ☐ Execute the script to verify success

☐ Exercise 5.3 – Shared OR, Update From Local:

- ☐ Open the Object Repository Manager
- ☐ Save the OR as “Flight”
- ☐ In UFT, associate the “Flight” OR to your test

  From the menu: Resources->Associate Repositories

  Don’t forget to associate Action 1.

- ☐ In the OR Manager, update the “Flight” OR with objects from script “OR – Parameterize”

  Note that the script shouldn’t be open in UFT at this time

  From the menu: Tools->Update From Local Repository

  Save the shared OR

- ☐ In UFT, open the “OR” script
- ☐ Open the OR

  View the objects in the shared and in the local OR

  Why aren’t the objects in the OR editable?

  _____

☐ Exercise 5.4 – Shared OR, Add Object:

- ☐ In the OR Manager, add the button "Delete Order" to the shared OR

  Use the "Add Objects" icon to point to the object

- ☐ In UFT, Open the OR

  Verify that the new object is in the OR

  Use the "Highlight in Application" icon to locate the object in the Flight application

☐ Exercise 5.5 – Object Spy:

- ☐ Open the existing script "Wrong OR"
- ☐ Run the script

  What data caused the script to fail? _____

  Why? _____

  What object caused the script to fail? _____

  Why? _____

- ☐ Fix the data issue
- ☐ Using the Object Spy, fix the Physical Description of the object in the OR

  If a property value of an object is different in the OR and the Object Spy, which value is the correct one?

  _____

- ☐ Run the script to verify its successful execution.

You have now completed Lab 5, Object Repository.

You may now continue with the next lesson in the Quick Test Professional course.

3. SYNCHRONIZATION

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 6: Synchronization

Lab Purpose

In this lab you will

- Add a synchronization point to a script

Check off each task below as you complete it:

☐ **Exercise 3.1 – No Synchronization:**

☐ Open the “Create Order” script

Is synchronization needed somewhere in the script?

_____

☐ Record a selection from the “Number of tickets” combo box at the end of the script (before pressing "New Search".

☐ In UFT:

Open menu: File -> Settings

Go to the “Run” tab

Change the “Object synchronization timeout” to 2 seconds.

☐ Run the script

Did the script pass or fail? Why? _____

_____

☐ Exercise 3.2 – Global Synchronization Time:

- ☐ In UFT:

  Open menu: File -> Settings

  Go to the "Run" tab

  Change the "Object synchronization timeout" to 20 seconds.

- ☐ Run the script.

  Did the script pass or fail? Why? _____

  _____

☐ Exercise 3.3 – Synchronization Point:

- ☐ In UFT:

  Open menu: File -> Settings

  Go to the "Run" tab

  Change the "Object synchronization timeout" to 2 seconds.

- ☐ Add a synchronization point at the required place

  Where did you enter the synchronization point?

  _____

  On which object and property did you synchronize?

  _____

- ☐ Run the script
- ☐ Is it a good idea to use the "Wait" statement as a synchronization point?

  _____

You have now completed Lab 3, Synchronization.

You may now continue with the next lesson in the Quick Test Professional course.

4. ALTERNATIVE RECORD TYPES

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 5: Alternative Record Types

Lab Purpose

In this lab you will

- Record a script in Normal, Insight and Analog modes.

Check off each task below as you complete it:

☐ **Exercise 2.1 – Insight Object:**

- ☐ Start a new script
- ☐ Record, in Normal mode, the flow of selecting the default flight

  Instead of clicking "Find Flights" click the "Order" button below it (if not recorded, learn the object manually and insert a line to your script)

  Which type of object did UFT learn? _____

- ☐ Run the script

  Was the "order" button clicked? How can you tell? If not, why?

  _____

  Change the button to either virtual object or insight object

  Which option is the correct one? Why? _____

- ☐ Save the script as “Record – Normal and Insight”

☐ Exercise 2.2 – Analog Recording:

- ☐ Start a new script
- ☐ Open MS Paint application

  Record a click on the Black button

  Paint something

  Do the same with the Yellow button

- ☐ Run your script

  Was the drawings reproduced? _____

  What will happen if color buttons will change places in next Paint version? _____

  Fix the script using Insight and Analog recording

- ☐ Save your script as “Record – Analog and Insight”

You have now completed Lab 2, Alternative Record Types.

You may now continue with the next lesson in the Quick Test Professional course.

5. CHECKPOINTS

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 7: Checkpoints

Lab Purpose

In this lab you will

- Add a Standard Checkpoint to a script
- Use a regular expression to create a generic checkpoint in a script

Check off each task below as you complete it:

☐ **Exercise 4.1 – Standard Checkpoint:**

- ☐ Open the “Create Order” script
- ☐ Add a Standard check point at the beginning of the script (after the Initial Condition row)

  To verify "London" and "Paris" are the default values in the “Fly From” and “Fly To” combo boxes

  What property are you verifying? _____
- ☐ Add a Standard check point to verify

  The "Order" button is disabled

  The "Order" button is enabled after inserting passenger name
- ☐ Run the script
- ☐ View the checkpoints results in the report

  How can you test the check point for failure? Suggest two ways.

  _____

☐ Exercise 4.2 – Standard Checkpoint – No Parameterization:

- ☐ Add a Standard check point at the end of the script after the click on the Insert Order button

  To verify the text in the "Order Completed" message

  Should the check point be located before or after the synchronization point? _____

- ☐ Run the script
- ☐ View the checkpoints results in the report

  Did the check point pass?

  _____

  If not, why?

  _____

  What do you need to do for this check point to work?

  _____

☐ Exercise 4.3 – Standard Checkpoint – Parameterization:

- ☐ Change the check point expected result to use a regular expression

  Suggest two options of regular expression.

  _____

- ☐ Run the script
- ☐ View the checkpoints results in the report

You have now completed Lab 4, Checkpoints.

You may now continue with the next lesson in the Quick Test Professional course.

6. Adding Programming Statements Without Recording

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 10: Adding Programming Statements Without Recording

Lab Purpose

In this lab you will

- Use the Step Generator

Check off each task below as you complete it:

☐ **Exercise 6.1 – Step Generator**

- ☐ Open a new script
- ☐ Record the “New Order” flow
- ☐ At the end of the script, use the “Step Generator” to:

  Use the "GetROProperty" function for the "nomOfTicketsCombo" field to retrieve its “text” property.

  Use an If – Else statement to check if the retrieved value is a number (use “IsNumeric” function)

  Use the “Reporter” object to report PASS for number and FAIL otherwise.

- ☐ Save your script as “My First Programming”

You have now completed Lab 6, Adding Programming Statements Without Recording.

You may now continue with the next lesson in the Quick Test Professional course.

7. PARAMETERIZING

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 11: Parameterizing

Lab Purpose

In this lab you will

- Add an input and an output parameter to a script
- Save values to a data table and to an environmental variable

Check off each task below as you complete it:

☐ **Exercise 7.1 – Input Parameter – Data Table**

- ☐ Start a new script
- ☐ Record the flow of Search Order
- ☐ Parameterize the following data to be retrieved from the data table, local sheet

  Radio button selection (Name or Order Number)

  The Name\Order Number accordingly

  What did you have to parameterize for the radio buttons selection?

  _____

- ☐ Execute your script
- ☐ Save your script as “Create Order - Parameters”

☐ **Exercise 7.2 – Output Parameter – Data Table**

- ☐ Add to your script

  Use an Output value to save the total price of the search result.

- ☐ Execute your script

Where can you see the saved price?

_____

☐ Exercise 7.3 – Output Parameter – Environment Variables

- ☐ Add two lines to your script:

  Use an output value to store the total price in Environment parameter

  Write this value to the data table.

- ☐ Execute your script
- ☐ Verify that the order number is written to the data table

You have now completed Lab 7, Parameterizing.

You may now continue with the next lesson in the Quick Test Professional course.

8. ACTIONS

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 12: Actions

Lab Purpose

In this lab you will

- Create reusable and non-reusable actions
- Rename an action
- Pass action parameters

Check off each task below as you complete it:

☐ **Exercise 8.1 – Reusable Actions, Create New Actions, Rename an Action**

☐ Start a new script

☐ Rename the existing action to “LogIn”

☐ Create two new reusable actions

Name them “NewOrder” and "LogOut"

☐ Record the appropriate activities for each action

In the “LogIn” action

Use SystemUtil.Run to invoke the Flight application

Ensure that the action invokes the application only if it’s not opened yet

Parameterize the “Name” field to use a value from the data table. Create at least two rows of data.

Which sheet should you use if you want each login to run one time for all the orders that the “NewOrder” action will run?

_____

In the “NewOrder” action

Parameterize the action to use values for the “Fly From” and “Fly To” combo boxes, from the data table. Create at least two rows of data.

Which sheet should you use if you want only this action to run few times?

_____

What will happen if you use the “Global” sheet for the “NewOrder” data?

_____

- ☐ Define the script and each action to “Run on all rows”
- ☐ Execute your script

  How many times did the script run?

  _____

  How many times did each action run?

  _____

- ☐ Save your script as “Actions”

☐ Exercise 8.2 – Reusable actions – Create new action, Pass action parameters

- ☐ In the “NewOrder” action

  Define an output action parameter of type “Number” Call it “OrderNumber”.

  Add a code to retrieve the order number from the application (Extract it from the full sentence) and assign it to the output action parameter.

- ☐ Create one new reusable action

  Name it “SearchOrder”

  Record the appropriate activity for this action

  Define the input action parameter of type “Number”. Call it “OrderNumber”

  Replace the appropriate place in the code to use this parameter.

  Set the action to run on all rows

- ☐ Save the script
- ☐ Open a new script. Save it as “Using Reusable Actions”

  Insert calls to the four reusable actions in the correct order (“SearchOrder” should come after “NewOrder”).

Send the order number received from “NewOrder” action to “SearchOrder” action

- ☐ Execute your script.

You have now completed Lab 8, Actions.

You may now continue with the next lesson in the Quick Test Professional course.

9. VBSCRIPT

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 13: VBScript

Lab Purpose

In this lab you will

- Add constants and variables to a script
- Add date and string functions to a script
- Use a “for loop” and an array in a script
- Create a custom synchronization point in a script
- Add a “do loop” to a script

Check off each task below as you complete it:

☐ **Exercise 9.1 – Constant and Variables, Conditions, Date and string functions**

☐ Start a new script. Save it as “VBScript”

☐ Record the process of “New Order”

☐ Make these additions to your script

Declare a public constant FLIGHT_APP and set its value to the Flight Application path.

Declare public variables: CurrentDate, NumberOfFromCities, arrFrom()

Using the Step Generator (or any other way), use SystemUtil.Run to invoke the flight Application at the beginning of the script

Use the constant FLIGHT_APP.

Add a condition to invoke the application only if it’s not open yet.

Assign the date into the CurrentDate variable, by using the VBScript Date() function

Use it as input value to “Date of Flight:” field, after adjusting it to the requested order and format. Hint: use Split\Right\Mid or other string manipulation functions.

- ☐ Execute your script

  Once when the Flight Application is closed

  Once when the Flight Application is already open

☐ Exercise 9.2 – For Loop, Arrays

- ☐ Make these additions at the end of your script

  Assign the number of items in the “From:” list into NumberOfFromCities variable (use the Step Generator to find the correct method)

  Use a “For” loop to write each value from the “From:” list into the arrFrom array. Dynamically increase the array size.

  What should be the initial value of i?

  _____

  What should be the upper limit of the loop?

  _____

  Use a “For” loop to run on all the array cells and write them to the report. Use the “Ubound” function as the upper limit of the loop.

- ☐ Execute your script and view the report.

☐ Exercise 9.3 – Do Loop - Custom Synchronization Point

- ☐ Make your synchronization point a comment
- ☐ Instead, create your own synchronization point

  Create a “Do – Loop While” loop that:

  Retrieves the "Enabled" value from “numOfTicketsCombo” combo box

  Keeps looping while this value is False

- ☐ Execute your script to verify correct synchronization.

☐ Exercise 9.4 – Verify Calculation using GetROProperty

- ☐ Open the “Create Order” script. Save it as “Custom Checkpoint”

- ☐ At the end of the script, add a verification to ensure that the value in Tickets combo box * value in Price field = value in Total field
  - Use If – Else statement
  - Use GetROProperty to get the values from the combo box and labels
  - Use the Mid function to remove the $ sign
  - Suggest other functions to use for this task _____
  - Use Cdbl function to change the values from string to number
  - Use breakpoints, watch variables, etc. to debug your script
  - Report the success or failure to the report

You have now completed Lab 9, VBScript.

You may now continue with the next lesson in the Quick Test Professional course.

10. PROCEDURES

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 14: Procedures

Lab Purpose

In this lab you will

- Write user defined functions
- Add a function to a library
- Associate a function library to a script

Check off each task below as you complete it:

☐ **Exercise 10.1 – Functions**

☐ Start a new script. Save it as “Functions”.

☐ Create a function called “fDivide” that accepts two numbers, divides the first number by the second number, and returns the result.

At the beginning of the function, check that the parameters values are legal. If not, return a False instead of the division result.

Write the returned value to the report.

Check the function for legal and illegal parameters values.

☐ Create a function called “fDivideA” that accepts two numbers, divides the first number by the second number, and returns the result by a ByRef parameter

At the beginning of the function, check that the parameters values are legal. If not, return False, else return True.

Write the returned value to the report.

Check the function for legal and illegal parameters values.

☐ Create a function called “fLogIn”

The function should contain the flow of Login

The function should return True for success and False for failure

How many parameters should the function accept? Suggest parameters names?

_____

☐ Create a function called “fCreateOrder”

The function should contain the flow of creating an order

How many parameters should the function accept? _____

How can you use dynamic values in the function without passing them as parameters? _____

The code should support cases where empty value is passed for parameter – meaning that the default value in the field shouldn't change

The function should return the order number in case of success and -1 in case of failure. Use a string function to isolate the order number and verify if it is a number

Which string function did you use to extract the order number?

_____

Is "Mid" function good for this task? _____

☐ Exercise 10.2 – Associate a function library to a script

☐ Open a new “Function Library”. Save it as “Functions Library”.

Cut fLogIn and fCreateOrder from the script and paste then into the function library.

Execute “Functions” script

Call both functions in the correct order

Verify return values and exit the script in case of returned False

Did the script pass?

_____

If not, why?

_____

Associate the function library to the script.

Execute the script to verify that the functions are working.

You have now completed Lab 10, Procedures.

You may now continue with the next lesson in the Quick Test Professional course.

11. DYNAMIC OBJECTS

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 16: Dynamic Objects

Lab Purpose

In this lab you will

- Set a dynamic value of an object in the Object Repository
- Use Descriptive Programming to set a dynamic value of an object that isn't in the Object Repository

Check off each task below as you complete it:

☐ **Exercise 11.1 – Verify Calculation using GetROProperty**

- ☐ Open UFT with the Web Add In
- ☐ Open the "UFT Web Demo.html" application
- ☐ Create a script to check\uncheck all checkboxes in the screen in the following ways:

  Use the Description object

  Use Descriptive programming. Hint: use a while loop and the exit method.

  Use SetTOProperty method. Hint: learn one checkbox into the OR. Identify the property that needs to be changed dynamically.

- ☐ Save your script as "Dynamic Objects"

You have now completed Lab 12, Dynamic Objects.

You may now continue with the next lesson in the Quick Test Professional course.

12. DB CHECKPOINTS

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 17: DB Checkpoints

Lab Purpose

In this lab you will

- Insert a DB Checkpoint into a script
- Parameterize a DB Checkpoint

Check off each task below as you complete it:

☐ **Exercise 12.1 – DB Checkpoint**

☐ Start a new test. Save it as “DB Check”

☐ Record the flow of “New Order”.

☐ Insert a DB check point to verify that the new order is in the DB

Where should the checkpoint be located?

_____

Application DB file is located under:
C:\Users\User\AppData\Local\UFT\Demo\DB

In the first screen of the wizard, check the “Maximum number of rows” check bow, and set the value to “1”.

The query should only check the value of the last “OrderNumber” field in “Orders” table. Sort it by “OrderNumber”. Should the sort be Ascending or Descending? Query: "select OrderNumber from Orders order by OrderNumber ____

- ☐ Execute your script

  Did the script pass? If not, why?

  _____

  What can you do to fix the problem?

  _____

☐ Exercise 12.2 – Parameterize DB Checkpoint Expected Result

- ☐ Add an Output value to your script to save the order number to the DataTable.
- ☐ Access the check point properties. You can do it by Right Clicking the "CheckPoint" in your script and selecting "Checkpoint Properties…"

  Parameterize the expected result to take the value from the same column you wrote the Output value to.

- ☐ Execute your script to verify that the DB checkpoint passes.

You have now completed Lab 13, DB Checkpoints.

You may now continue with the next lesson in the Quick Test Professional course.

13. HANDLING EXCEPTIONS

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 18: Exceptions

Lab Purpose

In this lab you will

- Create a recovery scenario and add it to a script
- Handle exceptions programmatically using Err objects

Check off each task below as you complete it:

☐ **Exercise 13.1 – Recovery Scenario Manager**

- ☐ Start a new test. Save it as “Recovery”
- ☐ Associate the “Functions Library” library to the script
- ☐ Record the flow of creating a new order
- ☐ Add a recovery scenario

    Select “Test run error” as the Trigger event.

    Select “Item in list or menu not found” as the Test run error.

    Select “Function call” as the Recovery operation.

    Browse for “Functions Library” library

        Define a new function

        This function will instruct UFT to select the first item in the list in case the recovery is activated.

        Write in the function

            Object.Select 0

            Report to the report that a default value was selected

    Select “Proceed to next step” as the Post-Recovery Test Run Options

    Save the scenario and associate with the test

- ☐ Execute your script

  Check the report to see if the Recovery Scenario was activated.

  Was the Recovery Scenario activated?

  _____

  If not, why?

  _____

  What do you need to do in order for the scenario to be activated?

  _____

- ☐ Change the value your script selects from "Fly From:" list to “Denverrrr”
- ☐ Execute your script

  Check the report to see if the Recovery Scenario was activated.

  Was the Recovery Scenario activated?

  _____

  If so, why?

  _____

☐ Exercise 13.2 – Err Object

- ☐ Start a new test. Save it as “VBScript Recovery”
- ☐ Associate the “Functions Library” library to the script
- ☐ Open the “Functions Library” library
- ☐ Copy and save the “fDivide” function as “fDivideB”.
- ☐ Remove all the existing error handling (parameters legal check)
- ☐ Use the Err object to handle errors in the function

  Use On Error Resume Next at the beginning of the function

  Use On Error GoTo 0 at the end of the function

  Check if Err.Number is different than 0 in any place an error may occur.

  If an error occurs, report Err.Description to the report and return False.

- ☐ Add a call to “fDivideB” to your script
- ☐ Execute the script

  Call the function with legal values

  Call the function with illegal values

- ☐ Check the report to verify that the error was reported

You have now completed Lab 14, Handling Exceptions.

You may now continue with the next lesson in the Quick Test Professional course.

14. DYNAMIC DATA

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous lab/labs and their prerequisite lesson(s).
- Lesson 19: Dynamic Data

Lab Purpose

In this lab you will

- Retrieve information from a data table
- Retrieve information from a combo box and write it to a data table

Check off each task below as you complete it:

☐ **Exercise 14.1 – Get data from Web table, ChildItem, SetTOProperty**

- ☐ Open UFT with the Web Add In
- ☐ Open the "UFT Web Demo.html" application
- ☐ Create a script to search the Reports table and check a check box

  Search the second column for value "andream_79473_01_2"

  Check the corresponding check box using "ChildItem" method

  Check the corresponding check box using "SetTOProperty" method
- ☐ Save your script as "Dynamic Objects 1"

☐ **Exercise 14.2 – Get data from Web Combo Box, Write to Data Table**

- ☐ Create a script to retrieve all environments from the "Env #:" Combo Box

  Retrieve all items from the combo box and write them to the local sheet (first create a column named: ENV) of the data table. Hint: Use a loop and the correct data table method to move to the next row.
- ☐ Save your script as "Dynamic Objects 2"

You have now completed Lab 15, Dynamic Data.

You may now continue with the next lesson in the Quick Test Professional course.

15. EXTERNAL DATA

Prerequisites

Before proceeding with this lab, make sure that you have completed the following:

- Previous labs and their prerequisite lesson(s).
- Lesson 20: External Data

Lab Purpose

In this lab you will

- Export values to a data table
- Import values from a data table
- Use database objects to retrieve values from a data table
- Use a file system object to export values to a text file

Check off each task below as you complete it:

☐ **Exercise 15.1 – Data Table Import and Export Methods**

☐ Create two new scripts (or use scripts from previous lessons). Save them as "New Order – Export" and "Search Order – Import". Record the appropriate activity for each script.

☐ In "New Order – Export"

Save the order number into the data table

Use either the "Export" or "ExportSheet" of the data table object to export to c:\Flight.xls.

Where should the "Export" command be located?

_____

☐ In "Open Order – Import"

Use either the "Import" or "ImportSheet" of the data table object to import from c:\Flight.xls.

Use the order number from the table as the order to open

Where should the "Import" command be located?

_____

☐ Verify successful execution

☐ Exercise 15.2 – Using Database Objects

- ☐ Create a new script. Save it as “External Data - DB”
- ☐ Record the flow of “Search Order”
- ☐ Using ADODB.Connection and ADODB.Recordset objects, connect to the Flight Application DB to retrieve the OrderNumber values from Orders table

  Use connection string: “Driver={SQLite3 ODBC Driver};Database=C:\Users\User\AppData\Local\UFT\Demo\DB\Flights.s3db;StepAPI=0;SyncPragma=;NoTXN=0;Timeout=;ShortNames=0;LongNames=0;NoCreat=0;NoWCHAR=0;FKSupport=0;JournalMode=;OEMCP=0;LoadExt=;BigInt=0;JDConv=0;PWD=”

  Use query: "select OrderNumber from Orders where OrderNumber < 6 order by OrderNumber Desc"
- ☐ Program your script to run in a loop and use the first five orders from the record set.
- ☐ Execute your script.

☐ Exercise 15.3 – Using File System Objects

- ☐ Save “External Data - DB” as “External Data – Text File”
- ☐ Delete the DB parts
- ☐ Prepare a text file (Notepad) with five rows. In each row, write the number of an existing order. Save it under C:\Orders.txt.
- ☐ Using the FileSystemObject object

  Program your script to run in a loop and use the five orders from the file.
- ☐ Execute your script.

You have now completed Lab 16, External Data.

You may now continue with the next lesson in the Quick Test Professional course.