



Open2Test Mainframe Test Automation Framework for QTP – Scripting Standards

Version 1.0

June 2011

DISCLAIMER

Verbatim copying and distribution of this entire article are permitted worldwide, without royalty, in any medium, provided this notice is preserved.

TABLE OF CONTENTS

| | |
|--|-----------|
| 1. INTRODUCTION | 3 |
| 1.1. Purpose | 3 |
| 2. STANDARDS FOR KEYWORD SCRIPTING..... | 4 |
| 2.1. Getting Started | 4 |
| 2.2. Column Description | 4 |
| 2.2.1. Automate (Column 'A') | 5 |
| 2.2.2. Action (Column 'B') | 5 |
| 2.2.3. Object (Column 'C') | 8 |
| 2.2.4. ActionValue1 (Column 'D') | 9 |
| 2.2.5. Comments (Column 'E') | 15 |
| 2.2.6. Delimiters..... | 15 |
| 3. SEQUENCE OF KEYWORDS..... | 17 |
| 3.1. Use of the Keyword 'Context' | 17 |
| 3.2. Use of 'Conditional Statements' | 18 |

1. Introduction

1.1. Purpose

This document provides details about the various columns used during scripting, the keywords and their descriptions, along with methodologies to be followed while scripting using keywords.

2. Standards for Keyword Scripting

2.1. Getting Started

Before learning about the columns used for keyword scripting, the user should know the keyword script and how to call the framework from the test script.

As shown in the figure below, the keyword script is the actual automation test script that corresponds to the manual test case. It is written in the global sheet of the tool. In the 'Expert View' of the tool, the framework is called using the command 'Call Keyword_Driver()' or Keyword_Driver.

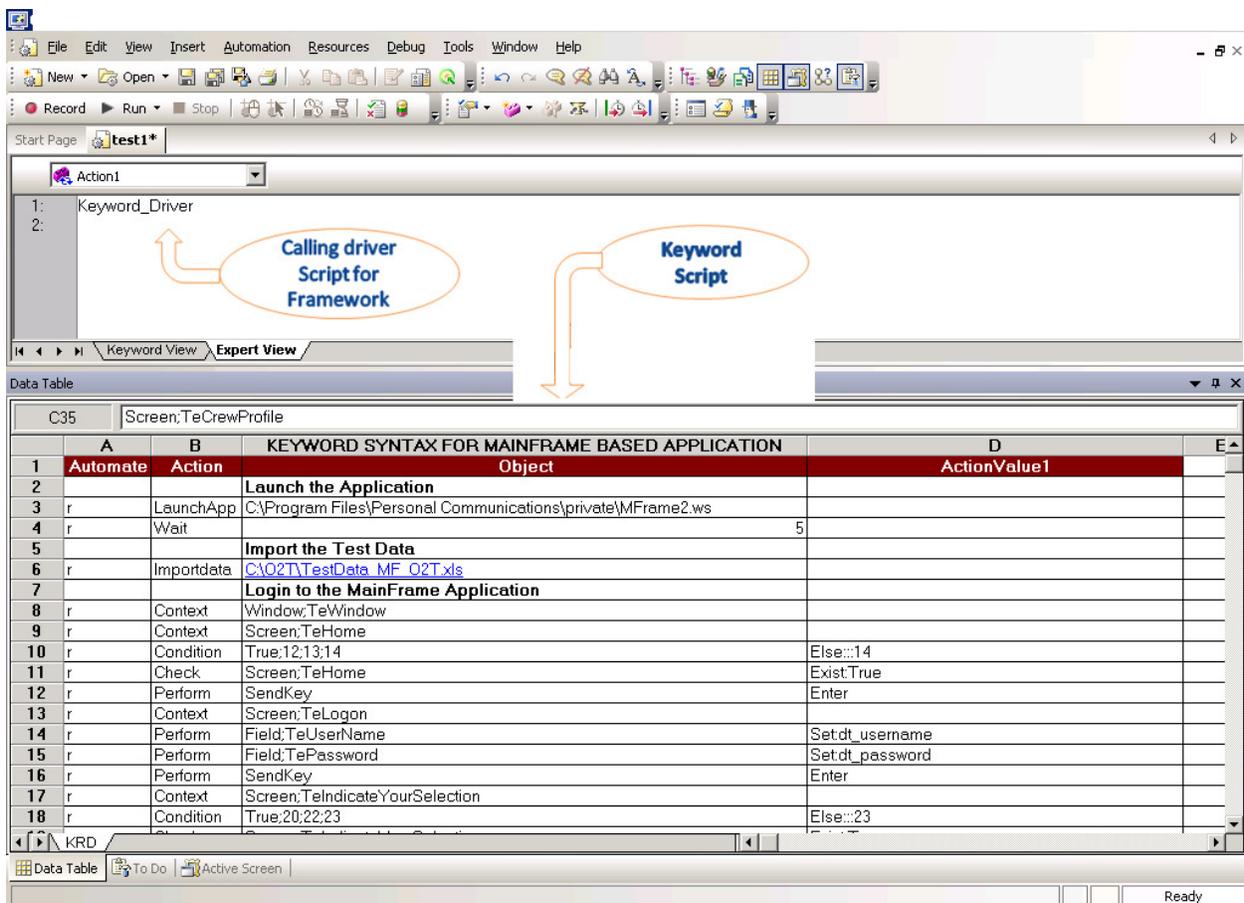


Figure 1: Keyword Script and Calling the Framework

2.2. Column Description

This section describes the columns used for keyword scripting.

2.2.1. Automate (Column 'A')

The data in the 'Automate' column identifies whether the current step in the test case is to be run (automated) or not. This column has the letter 'r', which denotes that the current step should be run. If any step in the test case is not being run then the corresponding row in the first column is to be left blank. All the steps will run based on the data in this column.

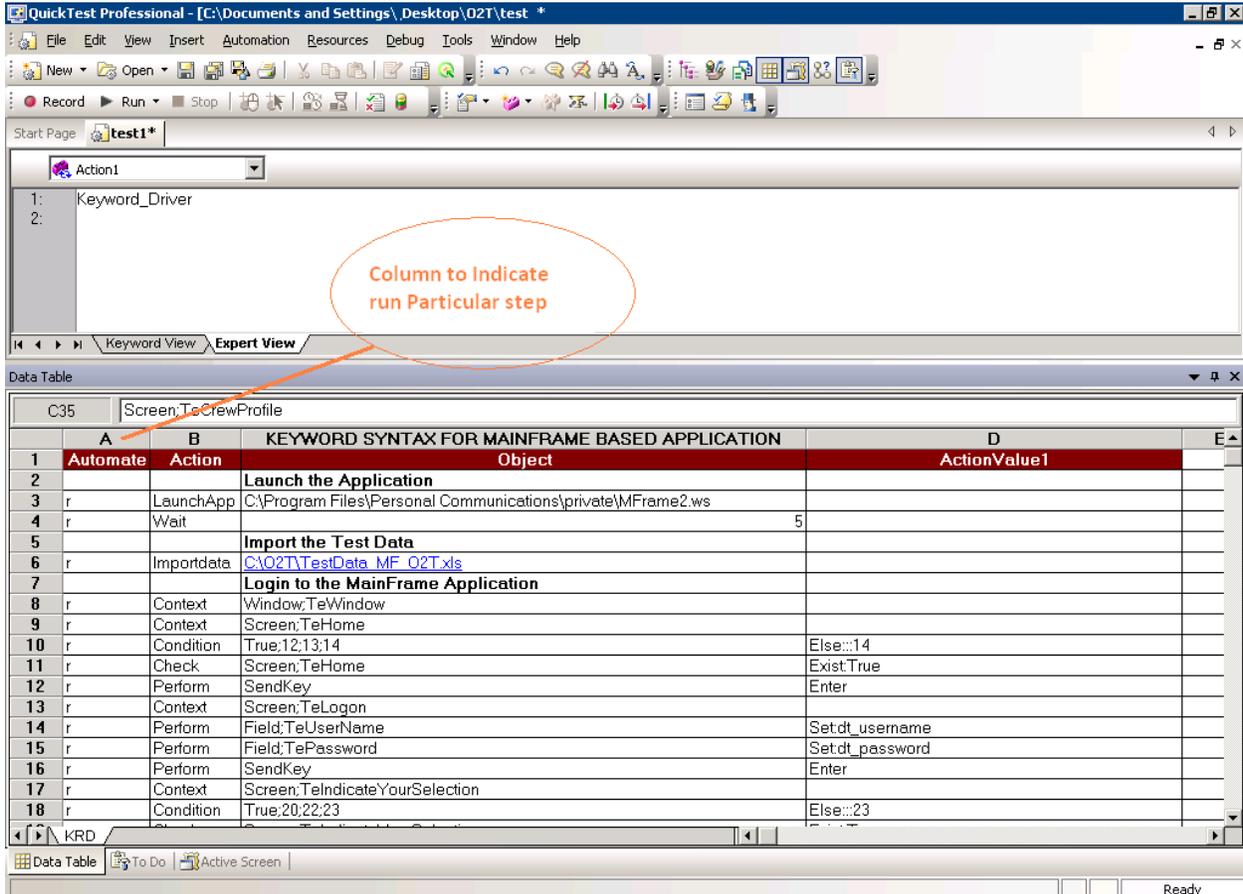


Figure 2: Column 'Automate'

2.2.2. Action (Column 'B')

The second column of the global sheet indicates the generic type of action being performed on the application under test (AUT). The Action column is dedicated to different types of actions that will be performed on a particular object.

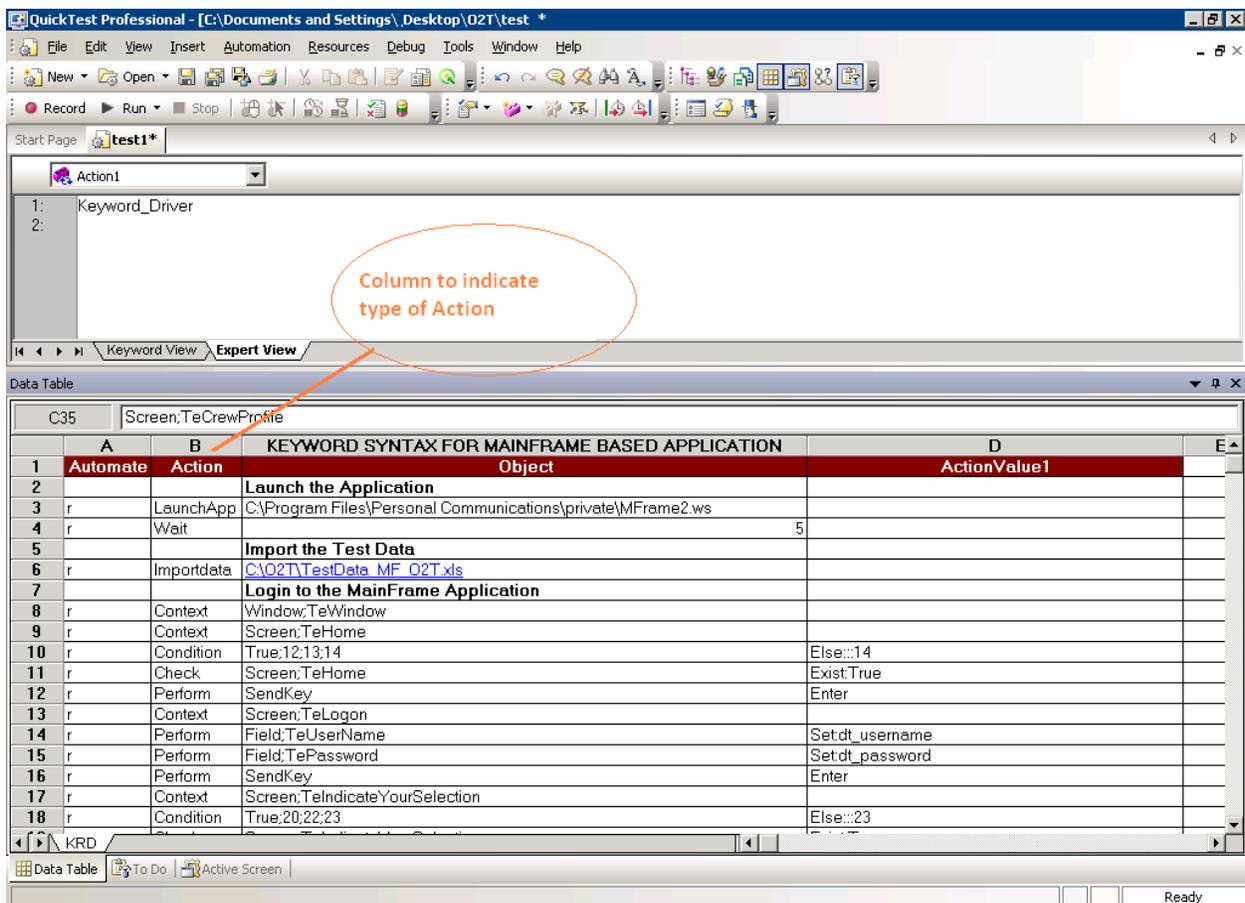


Figure 3: Column 'Action'

The keywords that can be used in this column are:

1. LaunchApp

'LaunchApp' is used to launch the AUT. This keyword triggers the driver script to launch the application either from a specified folder (the location of which is specified in the third column) or, if the application is already synchronized with QuickTest Professional (QTP), then this automatically launches the application from the location specified in QTP.

2. Context

'Context' is used only on a Window or a screen object. This keyword brings a particular Window or a screen object to the current context, so that any operation or checking can be performed on that particular Screen or window.

3. Perform

'Perform' is used to perform an operation on a particular object (ex: Passing a Keys like Enter or Function Keys, closing an application, or typing some text in a textfield). This

keyword should be entered in the corresponding row in the second column if any such operations are to be performed.

4. Check

'Check' is used to check if the required property of a particular object is attained at runtime. This is a type of validation step (expected result).

5. Condition

'Condition' is used to compare two variables, check properties for the existence of windows, etc.

6. CallFunction

'Call Function' is used to call any declared function used in a particular script. These functions should be declared in a different .vbs file.

7. Storevalue

'Storevalue' is used to store the property values of different objects in different environment variables. These environment variables can later be used as input parameters in various functions and also in scripts.

8. SendKey

'SendKey' is used to pass hot keys such as Enter, F3, F10, Ctrl-S, etc.

9. MsgBox

'Msgbox' is used for debugging to display the contents of a variable.

10. Report

'Report' is used for customized reporter events. It is displayed in the result sheet. The report results can include: i) Pass, ii) Fail, iii) Done, iv) Warning.

11. Strsearch

'Strsearch' is used to search for a 'sub string' inside a 'main string'.

12. Strreplace

'Strreplace' is used for replacing a 'sub string' inside a 'main string' with a new 'sub string'.

13. Strconcat

'Strconcat' is used to concatenate any number of strings with one other.

14. Wait

'Wait' is used to place static waits in the keyword script.

15. Arith

'Arith' is used to perform arithmetic operations on the variables.

16. Callaction

'Callaction' is used to call reusable actions that are declared in the script.

17. Loop

'Loop' is used to loop a set of actions given in the datatable.



A detailed description of the keywords is provided in the QTP Open Source Test Automation Framework Keywords for Mainframe document.

2.2.3. Object (Column 'C')

The third column of the global sheet is used to indicate the object on which a particular type of action is to be performed. When the object is present in the object repository, the object class and object name are specified in column C (as shown in Example 1).

Example 1:

| Action | Object |
|---------|------------------|
| Perform | Field;TeUserName |
| Perform | Field;TePassword |



The object and its name are usually separated by a delimiter ';' as shown in the above example. (Delimiters will be covered later in this document).

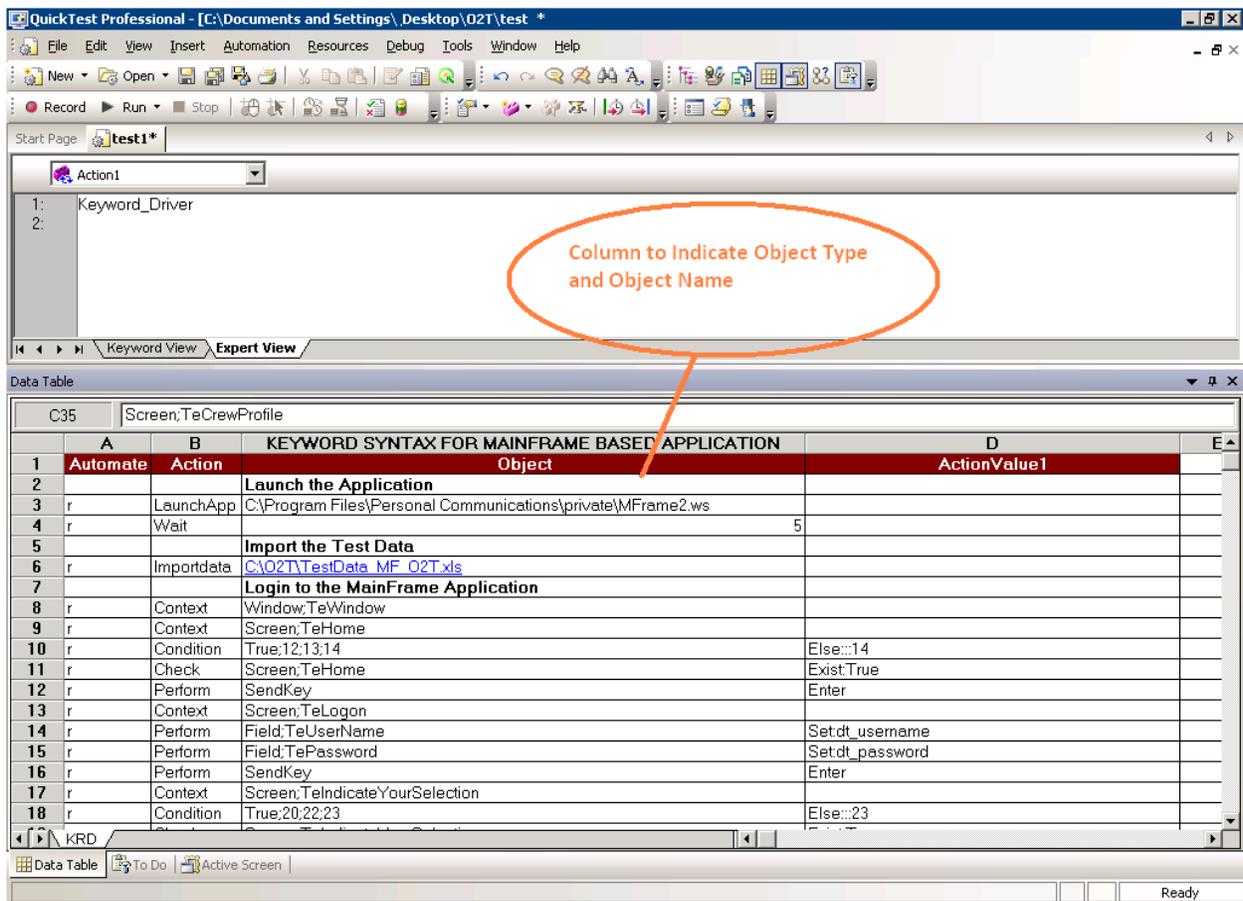


Figure 4: Column 'Object'

Commonly used objects include:-

| Sl.No | Objects used in the Open Source Test Automation Framework for Mainframes | Mainframes Object Class |
|-------|--|-------------------------|
| 1. | Window | TeWindow |
| 2. | Screen | TeScreen |
| 3. | Field | TeField |

Table 1: Objects used in the Open Source Test Automation Framework

2.2.4. ActionValue1 (Column 'D')

The fourth column of the global sheet is used to indicate the specific action being performed on the objects present in the AUT. It contains the details of all the operations or verifications to be performed on the objects listed in the objects column.

Consider the example of the object 'field' with name TeUserName.

One of the actions that can be performed on a Field would be enter value.

Therefore, in column 4 the above operation is put in the keyword form as **SET**.

Example 1: The keyword Set on a TeUserName field is as follows:

| Action | Object | Operation |
|---------|------------------|--------------|
| Perform | Field;TeUserName | Set:dt_uname |

ACTION



If user wants to **check** if the button is enabled before entering data, the syntax would be

| Action | Object | Operation |
|--------|------------------|------------|
| Check | Field;TeUserName | Exist:True |

CHECKING



Similarly, if the user wants to check whether the object is disabled or not available, the syntax would be

| Action | Object | Operation |
|--------|------------------|-------------|
| Check | Field;TeUserName | Exist:False |

CHECKING



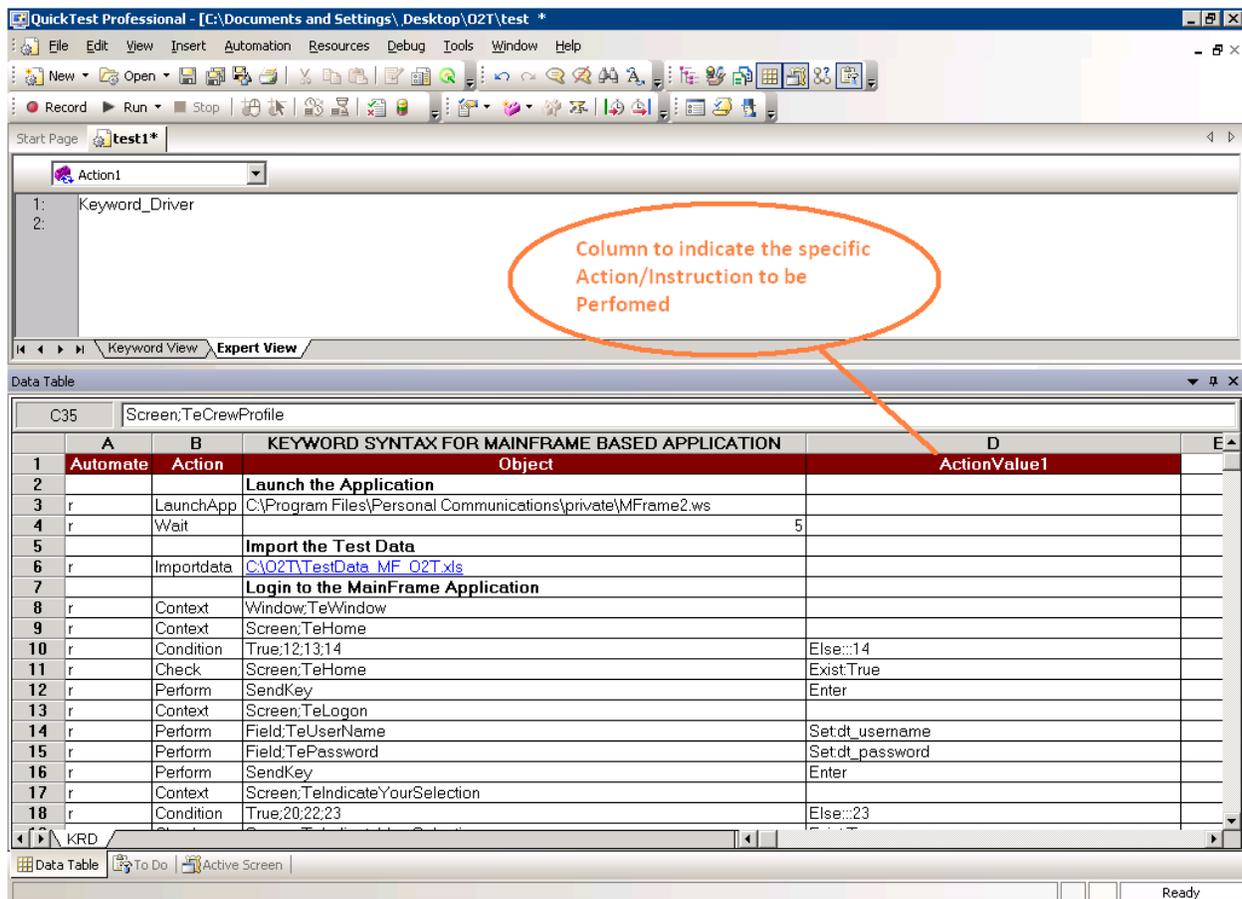


Figure 5: Column 'Actionvalue1'

The most commonly used keywords for specific actions that can be used with the generic keyword 'Perform' written in Column 'Action' are:-

1. **Activate**

Activates the window.

2. **Close**

Closes the window.

3. **PF1**

System will press 'PF1' key with the focus on the object that has current context.

4. **PF2**

System will press 'PF2' key with the focus on the object that has current context.

5. **PF3**

System will press 'PF3' key with the focus on the object that has current context.

6. **PF4**

System will press 'PF4' key with the focus on the object that has current context.

7. **PF5**
System will press 'PF5' key with the focus on the object that has current context.
8. **PF6**
System will press 'PF6' key with the focus on the object that has current context.
9. **PF7**
System will press 'PF7' key with the focus on the object that has current context.
10. **PF8**
System will press 'PF8' key with the focus on the object that has current context.
11. **PF9**
System will press 'PF9' key with the focus on the object that has current context.
12. **PF10**
System will press 'PF10' key with the focus on the object that has current context.
13. **PF11**
System will press 'PF11' key with the focus on the object that has current context.
14. **PF12**
System will press 'PF12' key with the focus on the object that has current context.
15. **PF13**
System will press 'PF14' key with the focus on the object that has current context.
16. **PF14**
System will press 'PF14' key with the focus on the object that has current context.
17. **PF15**
System will press 'PF15' key with the focus on the object that has current context.
18. **PF16**
System will press 'PF16' key with the focus on the object that has current context.
19. **PF17**
System will press 'PF17' key with the focus on the object that has current context.
20. **PF18**
System will press 'PF18' key with the focus on the object that has current context.
21. **PF19**
System will press 'PF19' key with the focus on the object that has current context.

22. **PF20**
System will press 'PF20' key with the focus on the object that has current context.
23. **PF21**
System will press 'PF21' key with the focus on the object that has current context.
24. **PF22**
System will press 'PF22' key with the focus on the object that has current context.
25. **PF23**
System will press 'PF23' key with the focus on the object that has current context.
26. **PF24**
System will press 'PF24' key with the focus on the object that has current context.
27. **TAB**
System will press 'TAB' key with the focus on the object that has current context.
28. **Enter**
System will press 'ENTER' key with the focus on the object that has current context.
29. **Set:<value>**
System will set the specified field with the given value.
30. **Set:dt_<column Name>**
Sets the value retrieved from Test Data sheet to the field object.
31. **Set:env<value>**
Sets the value retrieved from environment to the field object.
32. **Click<X-Coordinates,Y-coordinates>**
Cursor position will be set on screen based on X,Y coordinates.
33. **Descriptive Programming**
Descriptive programming can also used with the above object types when the object is not added to the object repository.

The most commonly used keywords for specific actions used with the generic keyword 'Check' written in Column 'Action' are:-

1. **Exist:<True/False>**
This is a Check operation that is used to verify whether the Web object whose name is specified exists or not.
2. **Text:<value to be checked for>**
Compares the actual text with expected text specified by the user
3. **text:dt_<datafieldname>**

Compares the actual text with expected text specified by the user.

The most commonly used keywords for specific actions used with the generic keyword '**Storevalue**' written in Column 'Action' are:-

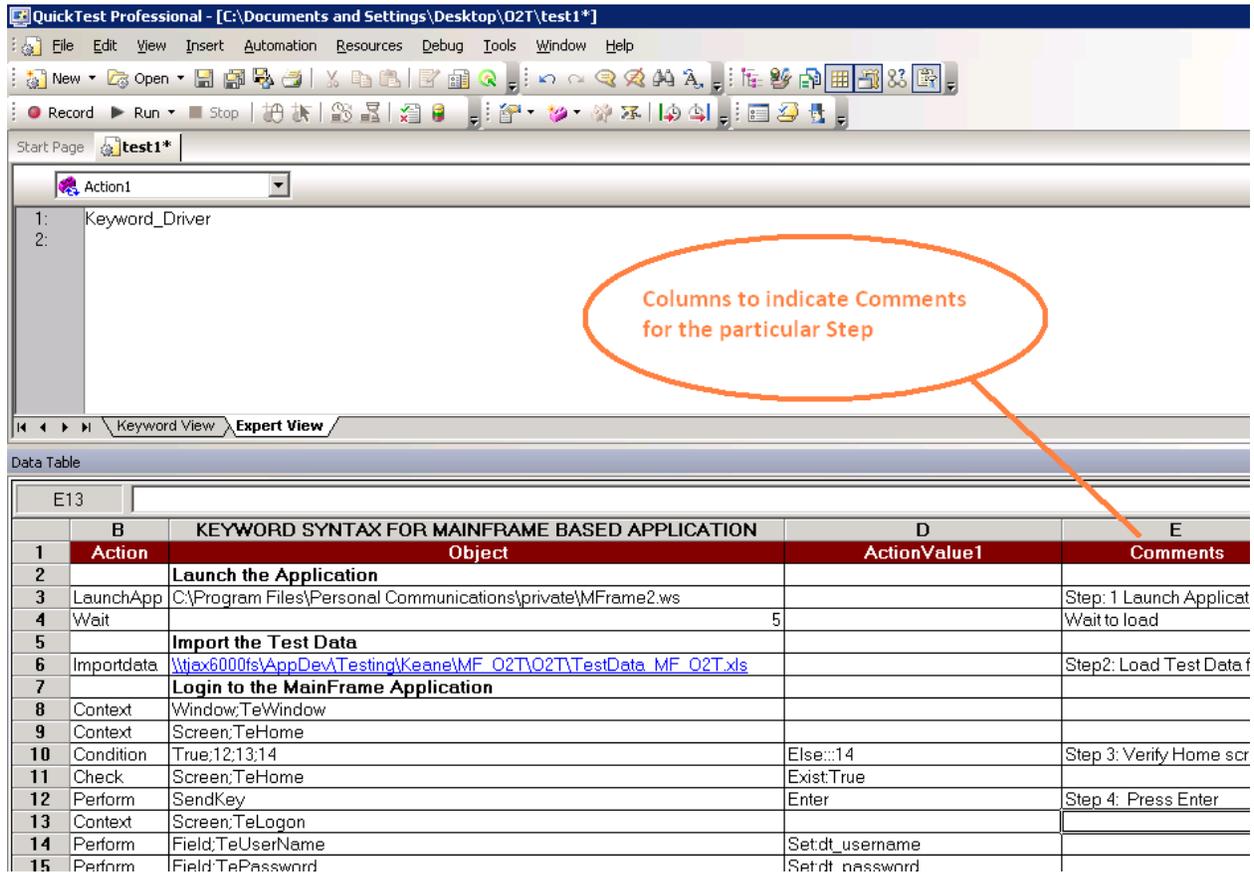
1. **Attached text:<var>**
Stores the text associated with the object in the variable provided by the user.
2. **fieldid:<var>**
Stores the value field id in the variable for the object.
3. **start column:<var>**
This is a Check operation that is used to verify the number of items in a Combobox.
4. **start row:<var>**
The user defined variable will have the value of start row based on the object position.
5. **columncount:<var>**
The user defined variable will have the value of total column count in the screen.
6. **current column:<var>**
The user defined variable will have the value of current column in the screen.
7. **current row:var**
The user defined variable will have the value of current row in the screen.
8. **inputfieldcount:var**
The user defined variable will have the value of input field count in the screen
9. **protected field count:var**
The user defined variable will have the value of protected field count in the screen.
10. **row count;<var>**
The user defined variable will have the value of row count in the screen.
11. **screenid:<var>**
The user defined variable will have the value of screen id of the screen object.
12. **text:<var>**
The user defined variable will have the whole text in the screen object.
13. **emulator status:<var>**
Stores the emulator status of window object in the variable.
14. **shortname:<var>**
Stores the shortname of window object in the variable.



A detailed description of the keywords is provided in the QTP Open Source Test Automation Framework Keywords for Web document.

2.2.5. Comments (Column 'E')

The 'Comments' Column is used to enter generic information about the current step being run. It provides a better understanding of the steps being performed in the particular test script and also helps to map the test script to the manual test case.



The screenshot shows the QuickTest Professional interface. The main area displays a test script with two steps: 'Keyword_Driver' and an empty step. Below the script is a 'Data Table' with 15 rows. The columns are labeled B, D, and E. Column B is 'Action', D is 'ActionValue1', and E is 'Comments'. An orange oval highlights the 'Comments' column, with a callout box containing the text 'Columns to indicate Comments for the particular Step'.

| | B | KEYWORD SYNTAX FOR MAINFRAME BASED APPLICATION | D | E |
|----|------------|--|----------------|-------------------------|
| 1 | Action | Object | ActionValue1 | Comments |
| 2 | | Launch the Application | | |
| 3 | LaunchApp | C:\Program Files\Personal Communications\private\MFrame2.ws | | Step: 1 Launch Applicat |
| 4 | Wait | | 5 | Wait to load |
| 5 | | Import the Test Data | | |
| 6 | Importdata | Wjex6000fs\AppDev\Testing\Keane\MF_O2T\O2T\TestData_MF_O2T.xls | | Step2: Load Test Data f |
| 7 | | Login to the MainFrame Application | | |
| 8 | Context | Window;TeWindow | | |
| 9 | Context | Screen;TeHome | | |
| 10 | Condition | True;12;13;14 | Else::14 | Step 3: Verify Home scr |
| 11 | Check | Screen;TeHome | Exist:True | |
| 12 | Perform | SendKey | Enter | Step 4: Press Enter |
| 13 | Context | Screen;TeLogon | | |
| 14 | Perform | Field;TeUserName | Setdt_username | |
| 15 | Perform | Field;TePassword | Setdt_password | |

Figure 6: Column 'Comments'

2.2.6. Delimiters

Delimiters are any string characters used to identify the sub-string limits. Delimiters are generally used with the Split function, which is used to split the input into different substrings. When a delimiter is omitted, the space character (" ") is assumed to be a delimiter.

Purpose of Delimiters:

Delimiters break down the input values to different strings and take them as keywords to perform any operation concerned with that object.

Delimiters Used in this Framework:

When scripting using the keyword-driven approach, use separators or delimiters appropriately between two keywords. Delimiters that are used in the framework are:

- : (colon)
- ; (semi colon)

Using Delimiters:

There are four columns involved in the keyword-driven approach. The role of delimiters comes in the 'Objects' column (column 3) and the 'Operations' column (column 4).

'Objects' column (column 3):

This column is used to define the class and the name of the object. The delimiter used in this column to separate the class of the object and the name of the object is a semi-colon ';'.

Example:

```
field; <fieldame>
```

'ActionValue1' column (column 4):

This column usually provides details of the operations that need to be performed on the object. The delimiter used to separate the property and the property values in this column is a colon ':'.

Example:

```
Set: <value>
```

3. Sequence of Keywords

While scripting using keywords, some keywords have to be written in combination with other keywords. This section deals with methodologies that should be followed while scripting using keywords.

3.1. Use of the Keyword 'Context'

The keyword 'Context' has to be used whenever the AUT screen changes. Example:

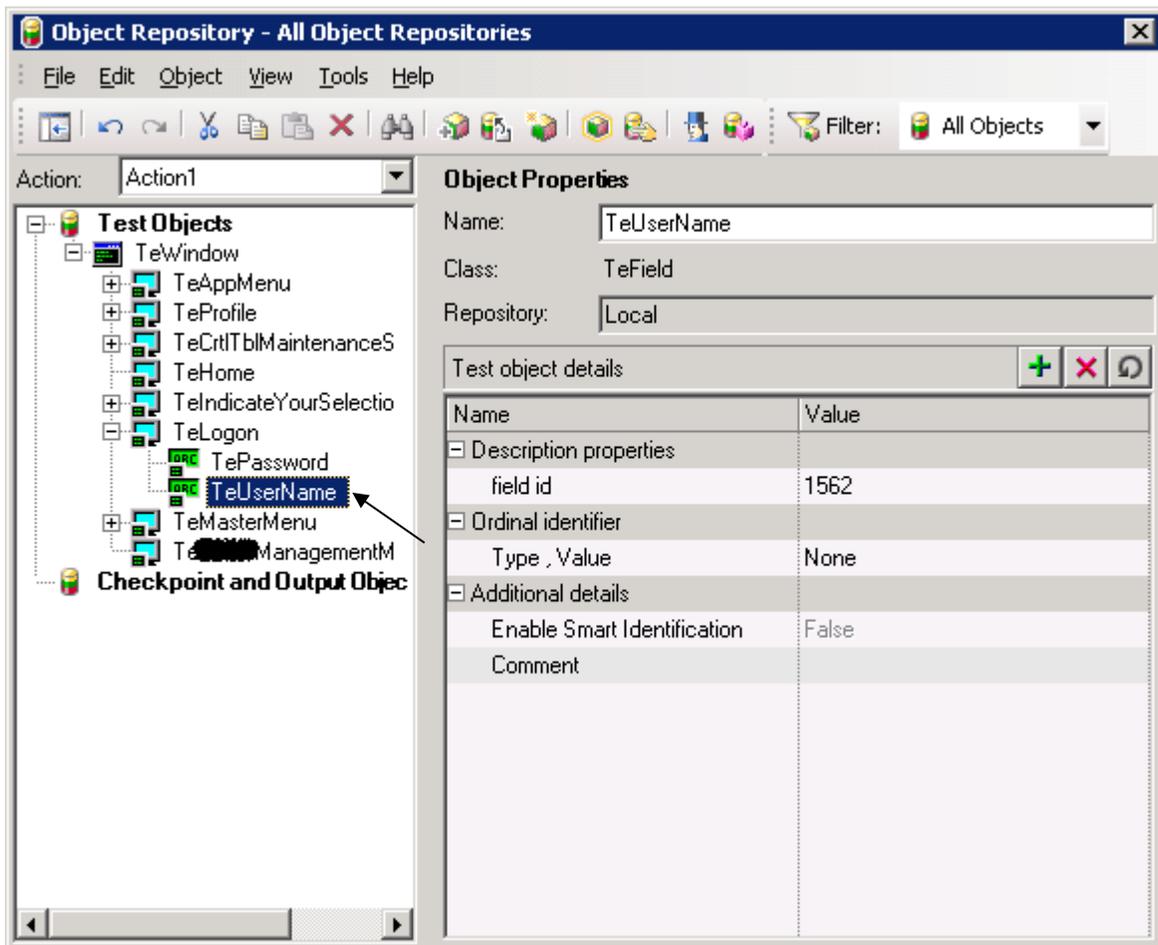


Figure 7: Keyword 'Context'

If the object 'TeUserName' has to be used in the script, then the preceding row should have the context set to the previous object in the hierarchy.

Therefore, the combination to be used while performing an action on the object 'TeUserName' is :

| | | |
|---------|------------------|-----------|
| Context | Window; TeWindow | |
| Context | Screen; TeLogon | |
| Perform | Field;TeUserName | Set:Smith |

If we are going to use another object on the same page, then the context need not be set again.

| | | |
|---------|------------------|-----------|
| Context | Window; TeWindow | |
| Context | Screen; TeLogon | |
| Perform | Field;TeUserName | Set:Smith |
| Perform | Field;TePassword | Set:Smith |

3.2. Use of ‘Conditional Statements’

If the user is implementing an If - Else conditional statement, then the keyword is followed by a semi-colon ‘;’ and the values that indicate the start row and the end row should be separated by a semi-colon ‘;’.

Example:

| | | |
|-----------|---------------------------------------|-------------------------------------|
| Condition | <True>;StartRow;endrow;resum eline | Else:startrow:endrow:resum eline |
|-----------|---------------------------------------|-------------------------------------|

If the condition mentioned is ‘True’, execution starts from the start row and would end at the end row specified and resumes from specified resume line or row. If the condition specified is ‘False’, there would be no effect in the script and the execution would continue as normal as mentioned in else statement.

COPYRIGHT

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.