Recent
Topics
Tutorials
Highlights
Settings
Feedback (http://community.
Sign Out

PREV
Overriding a Te
NEXT
l ov…

Settings
10 days left in your trial.
Subscribe.
Feedback
(http://community.safaribooksonline.cor
Sign Out

Registering a method to all classes

In some cases, we may need to customize a method that is common all TO classes, or at least to all references to a particular environmen such as the Web or Java. For example, suppose we need to customi: the `Click` method and register all Web TO classes. To write a statem for each class is quite tedious, and it may be an error-prone practice. For example:

```
RegisterUserFunc "WebEdit", "Click", "AQTP_Click"
RegisterUserFunc "WebButton", "Click", "AQTP_Click"
RegisterUserFunc "WebRadiobutton", "Click", "AQTP_Click"
```

If the new implemented method needs to be registered to all classes all environments, then it becomes impractical and undesirable to maintain this via code. We can manage such custom function registrations better by using a data-driven approach.

Getting ready

You can use the `FR_RegFunc.vbs` function library as in the *Overriding a Test Object method* recipe.

How to do it...

Perform the following steps:

1. In the function library, we will write the following code:

```
Dim QTP_TO_ENVS

Function LoadTOEnvironments()
        Set QTP_TO_ENVS = CreateObject("Scripting.Dictionary")
        QTP_TO_ENVS("Java") =     Array(    "JavaButton", _
                                        "JavaCalendar", _
                                        "JavaCheckBox", _
                                        "JavaDialog", _
                                        "JavaEdit", _
                                        "JavaExpandBar", _
                                        "JavaInternalFrame", _
                                        "JavaLink", _
                                        "JavaList", _
                                        "JavaMenu", _
                                        "JavaObject", _
                                        "JavaRadioButton", _
                                        "JavaSlider", _
                                        "JavaSpin", _
                                        "JavaStaticText", _
                                        "JavaTab", _
                                        "JavaTable", _
                                        "JavaToolbar", _
                                        "JavaTree", _
                                        "JavaWindow")
        QTP_TO_ENVS("Web") =     Array(  "Browser", _
                                        "Frame", _
                                        "Image", _
                                        "Link", _
                                        "Page", _
                                        "ViewLink", _
                                        "WebArea", _
                                        "WebButton", _
                                        "WebCheckBox", _
                                        "WebEdit", _
                                        "WebElement", _
                                        "WebFile", _
                                        "WebList", _
                                        "WebRadioGroup", _
                                        "WebTable", _
                                        "WebXML")

        QTP_TO_ENVS("StdWin") =     Array(  "Desktop", _
                                            "Dialog", _
                                            "Static", _
                                            "SystemUtil", _
                                            "WinButton", _
                                            "WinCheckBox", _
                                            "WinComboBox", _
                                            "Window", _
                                            "WinEdit", _
                                            "WinEditor", _
                                            "WinList", _
                                            "WinListView", _
                                            "WinMenu", _
                                            "WinObject", _
                                            "WinRadioButton", _
                                            "WinScrollBar", _
                                            "WinSpin", _
                                            "WinStatusBar", _
                                            "WinTab", _
                                            "WinToolbar", _
                                            "WinTreeView")
End function

Public Function ValidateTOClasses(ByVal strAddIn, ByRef arrTOClasse
    If IsArray(arrTOClasses) Then
        If UBound(arrTOClasses) = -1 Then
            'Empty Array, so assign default (all)
            arrTOClasses = QTP_TO_ENVS(strAddIn)
        End If
    Else
        'Not an Array, so assign default (all)
        arrTOClasses = QTP_TO_ENVS(strAddIn)
    End If
End Function


Function RegisterUserFuncEx(ByVal strAddIn, ByVal arrTOClasses, By
        Dim ix

        Call ValidateTOClasses(strAddIn, arrTOClasses)

        If IsEmpty(strOverriddenMethod) Or Trim(strOverriddenMetho

        For ix = 0 To UBound(arrTOClasses)
            If Left(arrTOClasses(ix), 1) <> "-" Then
                Call RegisterUserFunction(arrTOClasses(ix), strOve
            Else
                Print "Char '-' found at left position 1. Skipped
            End If
        Next
End Function

Function UnregisterUserFuncEx(ByVal strAddIn, ByVal arrTOClasses,
    Dim ix

    Call ValidateTOClasses(strAddIn, arrTOClasses)

    If IsEmpty(strOverriddenMethod) Or Trim(strOverriddenMethod) =

    For ix = 0 To UBound(arrTOClasses)
        If Left(arrTOClasses(ix), 1) <> "-" Then
            Call UnregisterUserFunc(arrTOClasses(ix), strOverridden
        Else
            Print "Char '-' found at left position 1. Skipped unre
        End If
    Next
End function
```

We will also add a custom function to override the `Click` method:

```
Function FR_Click(obj)
    Print "This is my custom Click function"

    obj.click
End Function
```

2. In `Action`, write the following code:

```
'Load the global dictionary with each environment's classes
call LoadTOEnvironments()
'Register the overriding method to all classes in StdWin env
call RegisterUserFuncEx("StdWin", array(), "click", "FR_Click")
'Register the overriding method to the WinEdit class
RegisterUserFunc "WinEdit", "Set", "FR_WinEdit_Set"
'Try to set the Agent Name field in the FR Login dialog
Dialog("Login").WinEdit("Agent Name").Set "mercury"
'Unregister the overriding method
UnregisterUserFunc "WinEdit", "Set"
call UnregisterUserFuncEx("StdWin", array(), "click")
```

In the output pane, you will notice that the printed script is **This is my custom Click function**, which is a line we added in our custom version of the `Click` method.

How it works...

Here, we will skip the parts of the code already explained in the *Overriding a Test Object method* recipe.

First, we call the `LoadTOEnvironments` function to get a dictionary object assigned to our `QTP_TO_ENVS` global variable, which will store for us a key for each environment we defined, each pointing to an array with a list of all the classes within that environment.

Then, we call the `RegisterUserFuncEx` function (our custom version of `RegisterUserFunc`) seeking to register the `FR_Click` function in the `Click` method for all classes (hence the empty array is passed) within the `StdWin` environment.

The ValidateTOClasses function checks whether the argument passed to the RegisterUserFuncEx function is an array or an empty array. In such a case, the function registers the custom method to all classes listed in the QTP_TO_ENVS(strAddIn) array. If a hyphen (-) is added to the left of the class name, then the method will not be registered to that particular class (and eventually, of course, will not be unregistered at all).

The flow is otherwise the same as in the previous recipe, but when the OK button of the pop-up dialog is clicked, UFT reroutes the call to our registered FR_Click function, and hence, the custom output (as mentioned in the previous section). In a way, we can say that our custom function is actually a delegate of the native method.

At the end of the script, we unregister the custom function from all classes in the environment (in this case, StdWin).

V
rriding a Tes…

NEXT
Using method ov…