



Advanced UFT 12 for Test Engineers Cookbook

Recent

Topics

Tutorials

Highlights

Settings

Feedback (<http://community.safaribooksonline.com>)

Sign Out

Settings

10 days left in your trial.

[Subscribe](#).

Feedback
(<http://community.safaribooksonline.com>)

Sign Out

PREV
Using child obj...

Aa



NEXT
obj...

Using native properties for object identification

With UFT, it is very straightforward to access identification properties using [GetROProperty](#) and [CheckProperty](#). However, this limits the scope of what can be checked, as TOs bring many properties that may be required. Fortunately, at least in web applications, we can also use native properties for object identification. In this recipe, we will demonstrate this feature using adapted examples from the previous recipes.

Getting ready

Please ensure that the Internet Explorer application is open on Google

How to do it...

We will identify the input query [WebEdit](#) object using a [Description](#) object. In [Action](#), we will put the following code:

```
Set desc=Description.Create
desc("attribute/nodeName").value="INPUT|input"
desc("attribute/nodeName").RegularExpression = true
desc("attribute/name").value="q"

set oQuery=Browser("title:=Google").Page("title:=Google").WebEdit(desc)

With oQuery
  If .exist(0) Then
    .highlight
  End If
End With
Set oQuery=Nothing
Set desc=Nothing
```

How it works...

Basically, this code works exactly as detailed in the *Using the Description object* recipe. The difference is in the way we defined the properties in the [Description](#) object; while in the previous example we used identification properties, here we use native or runtime properties. The [attribute](#) (or prefix) variable indicates that it is a runtime property. Recall that the standard windows Test Objects lack runtime properties and object capabilities. The name of the property must be valid.

There's more...

The reader may ask, when would using native properties be useful? There are several cases in which it would be useful, as follows:

- First, there are cases where none of the TO properties provided by UFT yield robust, consistent object identification. In other cases, these properties are not reliable due to possible glitches. In such a case, the [Description](#) object identifier might be the last resort and the only possibility to achieve unique object identification.
- Second, it enables us to get the value of a native property without the need to use the [Object](#) property. For example:

```
ImgProtocol=Browser("App Perform").Page("Application Performance").
```

- There are cases where we would need to refer to the [UniqueID](#) native attribute, which is not even available through the UFT Spy. This property is similar to `hWnd` we know from Windows

```
'Get the UniqueID for an identified Web element
imgID=Browser("Browser").Page("Page").Image("Image").GetRCProperty

'Use the UniqueID with an inline description
Browser("Browser").Page("Page").Image("attribute/UniqueID:="+imgID)
```

Refer to an article by Meir Bar-Tal at <http://www.advancedqtp.com/using-runtimeattributes-to-describe-qtp-web-objects/>
(<http://www.advancedqtp.com/using-runtimeattributes-to-describe-qtp-web-objects/>).

