



Advanced UFT 12 for Test Engineers Cookbook

Recent

Topics

Tutorials

Highlights

PREV
5. Object Identi

NEXT
ptiv...

Aa



Setting mandatory and assistive properties for a class

In this recipe, we will see how to change the default settings for object identification, which identifies the TO properties that UFT uses in its attempt to achieve unequivocal object identification. After reading this recipe, you will be able to view, analyze, and change the properties UFT uses to identify a TO. You will also acquire a deep understanding of how these properties are used by UFT to achieve object identification.

Getting ready

Before trying this recipe, ensure that the required hooks in the technology adapters, which are defined as add-ins or plug-ins for each of the relevant technologies (Web, .NET, Java, and so on.), are loaded and that the current test has them defined in its settings.

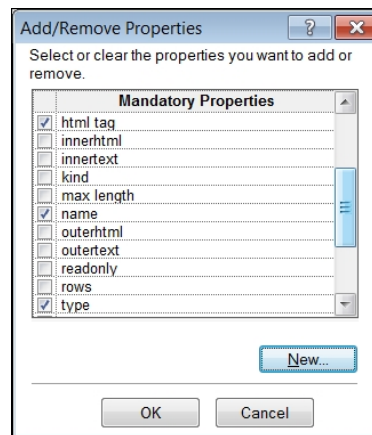
How to do it...

Proceed with the following steps:

1. From the UFT home page, navigate to **Tools | Object Identification**.... The following dialog will open:

The preceding screenshot shows the dialog after selecting **Web** from the **Environment** listbox and clicking on the **WebEdit** class. We can see that by default, UFT defines three mandatory properties (**html tag**, **name**, and **type**) for this class and has no assistive property.

2. We can change the set of **Mandatory Properties** by clicking on the **Add/Remove** button below the list. The following dialog will open:



We can then mark or unmark any of the listed properties. The properties we choose must have a solid logic behind them: one that will enhance the ability to identify TOs of the particular class correctly. For instance, if an ID is used instead of a name, then we may wish to use only the ID as the identifying property, as it

Settings

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

Sign Out

Settings

10 days left in your trial.

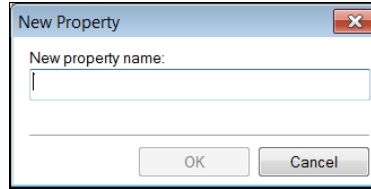
Subscribe.

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

Sign Out

should be unique for each web element.

3. The same procedure can be used for the **Assistive Properties** list by clicking on its **Add/Remove** button. Please note that mandatory and assistive properties are mutually exclusive.
4. Though it is a rarely used feature, it is possible to add custom properties to the default list of properties by using the **New** button, which opens the following dialog:

A screenshot of a 'New Property' dialog box. It has a title bar with the text 'New Property' and a close button (X). Inside the dialog, there is a label 'New property name:' followed by a text input field. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

You must ensure that the property is added by the developer to the objects of this class in the application code.

How it works...

Let us review how object identification is accomplished by UFT and define the basic concepts that underlie in this recipe.

By default, each TO class has two sets of properties defined, namely, mandatory and assistive. UFT mechanism for object identification works as follows:

When you add a TO to OR (for example, by recording a script or by using add object/define new object from the OR menu), the set of mandatory properties is used in full to identify the object, and these will appear in the **Description** properties section on the right-hand pane of the OR. If, however, more than one TO matches the same description as these mandatory properties, UFT will attempt to distinguish between them using the assistive properties. If none are defined for that particular class, then an ordinal identifier, *Location* or *Index*, is used. The *location* identifier refers to the relative position on the client screen, beginning from top to bottom and moving left to right. The *index* identifier refers to the order in which objects appear in the underlying code. For browser objects, UFT uses *creationtime* as the ordinal identifier, which is a zero-based index for browser objects, where the first browser gets value zero, the second one gets the value one, and so on.

Finally, if all the attempts fail, UFT will strive to use an identification mechanism called **Smart Identification (SI)**. This algorithm will try to find the best match for the given description, but it is highly recommended to disable this feature in the project settings and make every possible effort to ensure that all TOs are unequivocally identified. It is possible to disable the use of SI for each run session from the **Run** tab in the **Test Settings** dialog (select this by navigating to **File | Settings**).

There are two reasons for this recommendation. First, the SI mechanism is used as a last resort, so it will start only after UFT fails to identify the object based on the description. Therefore, it might affect the performance of the test. Second, there is no guarantee that the match found by SI is the one you intended to use, and this may lead to unexpected results.

