



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  
Resolving file lo

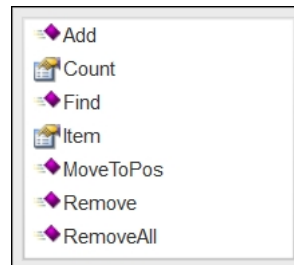
Aa



NEXT  
cre...

## Loading shared object repositories (RepositoriesCollection)

The `RepositoriesCollection` object provides methods to manage the Shared Object Repositories (SOR) associated with the actions of the test. The following screenshot shows the methods and properties of the `RepositoriesCollection` object:



As you might suspect, we can see a recurrent pattern again; the object is very similar to a Dictionary with methods such as `Add`, `Remove`, `RemoveAll`, and the properties `Count` and `Item`. All methods and properties operate in the context of the current action accordingly:

- The `Add` and `Remove` methods actually add and remove an SOR to and from the `RepositoriesCollection` object, respectively
- The `RemoveAll` method removes all SORs associated with the current action
- The `MoveToPos` method changes the priority order of an SOR (that is, its index in the collection)
- The `Find` method returns the index of a repository within the collection

## Getting ready

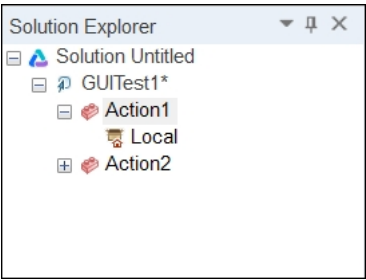
From the **File** menu, navigate to **New | Test**, or use the `Ctrl + N` shortcut. Open the Notepad application and add objects to an SOR (not the local OR) by using the Object Repository Manager. Save the SOR in a folder of your choice.

## How to do it...

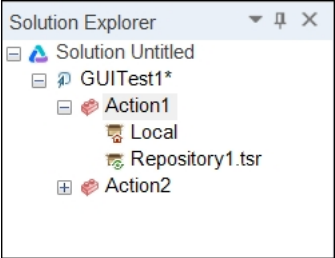
In the `Action` object of the test, write the following code (replace the path with your own):

```
RepositoriesCollection.Add "C:\Temp\Repository1.tsr"  
  
ExitTest
```

Before running the test, the solution explorer will look like the following screenshot, showing only the `Local` OR in the repositories collection:



Insert a breakpoint on the line with the `ExitTest` statement (select it and press `F9`) and then run the test. The run session will pause on that line. Now, watch the **Solution Explorer** window. It should look similar to this:



Notice the lead node with `Repository1.tsr` and also the SOR we wanted to load.

Now, press `F10` or `F5`, and watch the **Solution Explorer** window as the test ends. It is restored back to its initial state without the dynamically loaded SOR.

### How it works...

The `RepositoriesCollection` object is a kind of Dictionary that can load SORs dynamically and thus, associate them with the current action. The important point to make here refers to when such a feature would be useful, as opposed to the static association of SORs, which is a more common practice. The answer to this question is simple. If the requirements from the automation team/developer include supporting multiple versions of an application, then the `RepositoriesCollection` object would be very useful. For example, we may use an SOR for each version and load it dynamically at the start of the test according to the application version (which can be taken from an `Environment` variable or another source).

