Storing data in a DataTable

Sometimes, data that is collected during a run session might be need for later use. For example, suppose that **Application Under Test** (**AL** is a mobile operator management system. We could begin by executi a customer creation process, during which a customer ID is assigned automatically by the system. We then proceed with the other operations, such as selecting a phone number, an IMEI, credit card details, and so on. Later, we may wish to retrieve the customer recor and update some personal data such as the mailing address. For this purpose, we will keep the customer ID in the global datasheet, so that any action that is executed, which can be referenced later (for examp one that performs a customer search), will have access to the data.

> **Tip**
>
> Data stored in the global datasheet is effective only until the test stops. To see how to save data persistently for later run sessions, please refer to the *Exporting a DataTable* and *Importing an Excel file to a test* recipes.

How to do it...

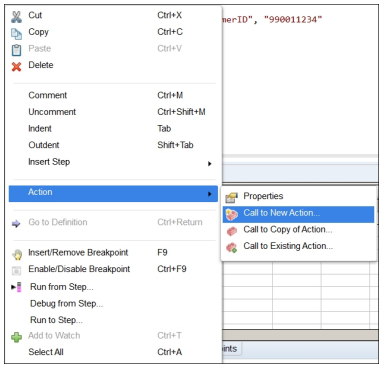
Proceed with the following steps:

1. From the **File** menu, select **New** | **Test** or use the *Ctrl + N* shortcut. When the new test dialog opens, choose **GUI Test** and click on the **Create** button.
2. We will save the value of a DataTable parameter, `CustomerID`, to the global sheet with the following code written in the code editor inside `Action1`:

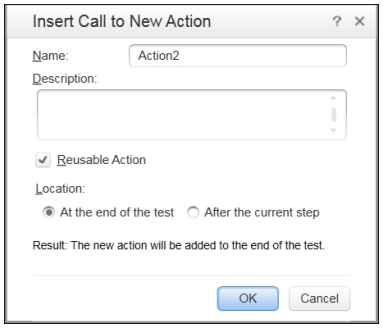
```
Dim CustomerID

DataTable.GlobalSheet.AddParameter "CustomerID", "990011234"
CustomerID = DataTable("CustomerID")
Print Environment("ActionName") & ": " & CustomerID
```

3. To retrieve the value from another action, we will now create a new action datasheet. In the code editor, right-click on the next empty line and select **Action** | **Call to New Action**, as shown in the following screenshot:



The following dialog will open:



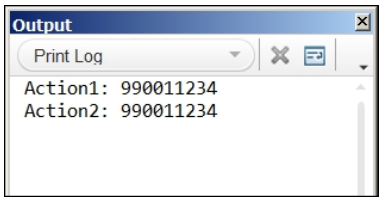
4. Leave all the fields with the default values and click on **OK**. You will see that a new action named `Action2` will appear in the **Solution Explorer** window and open on the code editor's MDI region:

5. Now, we will retrieve the value of the `CustomerID` parameter from the global sheet with the following code inside `Action2`:

```
Dim CustomerID

CustomerID = DataTable("CustomerID")
Print Environment("ActionName") & ": " & CustomerID
```

The result of this code in the UFT's console is shown in the following screenshot:



How it works...

When we run the test, UFT first executes `Action1`, and a new parameter named `CustomerID` will be added to `GlobalSheet` (a property of the `DataTable` object that refers to the `GlobalSheet` object) with the value given by the second parameter.

```
DataTable.GlobalSheet.AddParameter "CustomerID", "990011234"
```

We then immediately assign a variable with the retrieved value and print it to the console (for illustration purposes, we also concatenate the current action's name from the `Environment` object's built-in variables).

```
CustomerID = DataTable("CustomerID")
Print Environment("ActionName") & ": " & CustomerID
```

Next, UFT executes `Action2` same as `Action1`.

There's more...

There are other alternative ways of keeping and sharing data during a run session. The simplest is by using public variables declared in a function library attached to the test. The disadvantage of this approach is that these variables must be declared in advance and they are hard coded, but the nature of automation often demands more flexibility to manage such data.

See also

For information on advanced methods to share data among sessions, refer to the *Using a global dictionary for fast shared data access* recipe.

PREV
Retrieving data f…

NEXT
Importing an Exc…