

Local Datasheets How To

You can use QA Wizard Pro to test multiple scenarios and data ranges by using a local datasheet with your testing script and replacing data in the script with values from the datasheet. This How To walks you through creating a local datasheet, associating it with a script or using datasheet statements and functions, and running the script using datasheet values.

Local Datasheets Overview

You can test a variety of data with one script by creating a local datasheet in QA Wizard Pro. After the datasheet is associated with a script, the input values are read from the associated datasheet when the script runs. The script uses each row in the datasheet as it repeats.

For more advanced testing you can use datasheet statements and functions in scripts to navigate, read from, and write to local datasheets, and limit the data used in a datasheet. For example, you can use data from multiple datasheets or append rows to a datasheet when running a script.

Note: You do not associate a datasheet with a script when using datasheet statements and functions.

By using a local datasheet with a script, you can:

- Complete tests with a variety of data using one script
- Easily modify test data and the scripts that use it
- Maintain test data in QA Wizard Pro
- Test an application with dynamic user interface elements

Local Datasheets Scenario

This local datasheets scenario describes how to create a local datasheet, associate it with your testing scripts, assign datasheet values in your scripts, use datasheet statements and functions, and modify local datasheet properties.

Setup

Before you begin this local datasheets scenario, take the time to perform the following setup tasks:

- **Set up the testing environment**—Make sure you have access to the hardware, software, and network resources you need to record or run tests.
- **Create a workspace**—Workspaces organize scripts and related reports and datasheets. Depending on your organization's process, you may use one workspace for each application or one workspace for each functional area.
- **Set up the application in the application repository**—Application repositories store information about the tested application and version. Each version contains window and control data that identifies and locates objects.
- **Set general and playback options**—Take a few minutes to set general options that control how QA Wizard Pro works and set playback options that control how scripts run.

Local Datasheets Instructions

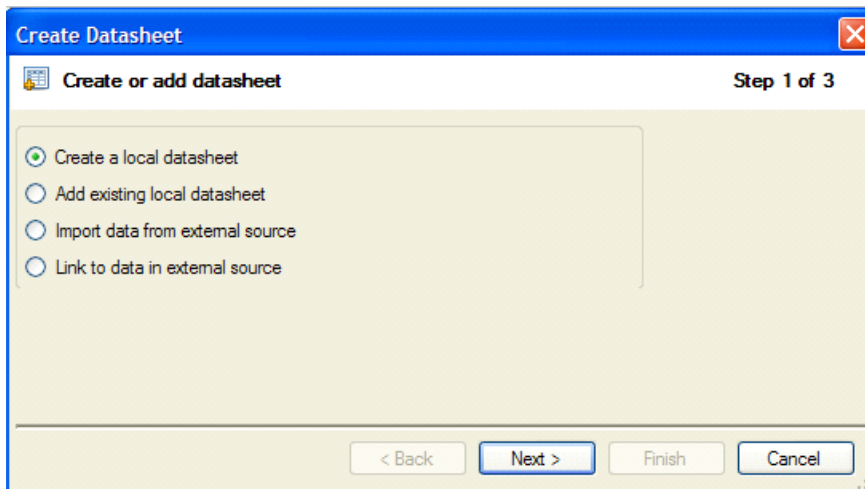
After you create a datasheet, you associate it with a script or add datasheet statements and functions to the script, depending on your testing scenario. When the script runs, the input values are read from the datasheet.

Creating local datasheets

Local datasheets are helpful if you have a small amount of data to test or you want to create and maintain test data in QA Wizard Pro.

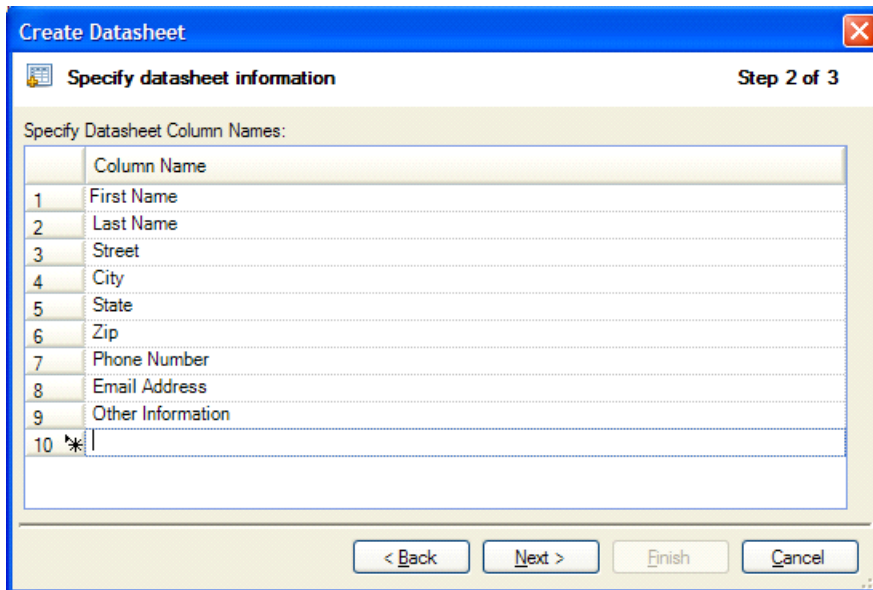
1. Choose **Data > Create Datasheet**.

The Create Datasheet wizard starts.



2. Select **Create a local datasheet** and click **Next**.
3. Enter a **Column Name** and press **Enter**.

Tip: Name the columns to match the fields in your user interface. For example, if you are creating a datasheet to test fields on a web form, make sure the column names match the name of the field names.



Tip: Right-click and choose Insert Row or Delete Row to add or remove a column name.

4. Click **Next** when you finish entering the column names.
5. Click **Browse** to select the Datasheet Path and name the datasheet.
6. Optionally enter a datasheet **Description**.
7. Click **Finish**.

The datasheet is saved and opens in the Script pane.

8. Click a cell to enter test data in the datasheet. Press **Tab** to move to another column. Press **Enter** to move to the first cell in the next row.
9. Choose **File > Save** to save the changes.

Adding existing local datasheets

If you want to use a local datasheet that is in another workspace, you can add it as an existing datasheet. When you add an existing datasheet, the datasheet file is shared and not copied. Any changes you make to the datasheet are simultaneously made in all workspaces.

1. Choose **Data > Create Datasheet**.
The Create Datasheet wizard starts.
2. Select **Add existing local datasheet** and click **Next**.
3. Click **Add**.
The Add Datasheet dialog box opens.
4. Select the local datasheet you want to add and click **Open**.
5. Click **Finish**.

Using local datasheets with scripts

After you create a datasheet, you can associate it with a script or add datasheet statements and functions to the script to perform more advanced testing.

Associate the datasheet with a script if:

- The script uses data from one datasheet.
- You want to run the entire script once for each datasheet row.
- You do not need to add or delete rows in the datasheet when the script runs.
- Called scripts are not associated with a datasheet or are associated only with the same datasheet as the main script.

Use datasheet statements and functions if:

- You only want to run the script once and use one datasheet row per run.
- You want to run a small portion of the script once for each datasheet row.
- You want to append a row to the datasheet when the script runs.
- The script uses data from more than one datasheet.
- The script uses an external datasheet set to use a script-generated query to retrieve data during playback.
- Called scripts are associated with different datasheets than the main script. QA Wizard Pro only uses the datasheet associated with the main script during playback. The main script fails if called scripts use different datasheets.

Note: You cannot use datasheet statements and functions if the datasheet is associated with a script.

Associating datasheets with scripts

When a datasheet is associated with a script, the script runs once using the first datasheet row and then repeats for each additional row until it reaches the end of the datasheet. You can associate a datasheet with a script from the Script pane or script properties.

Associating datasheets from the Script pane

1. Open the script you want to associate the datasheet with.
2. Select the datasheet from the **Data Source** list in the Script pane.

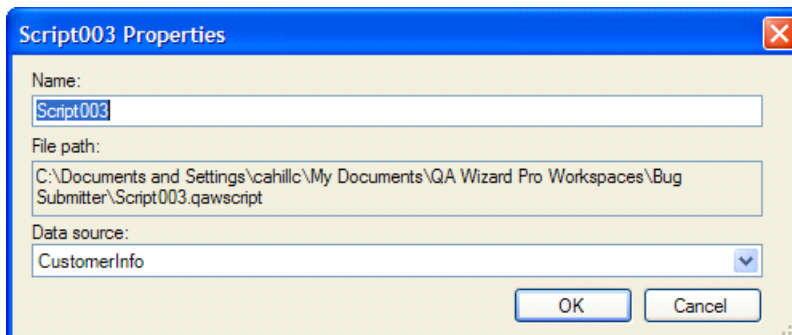
3.

#	Action	Information	Control	Window	Comment
0	Main				
1	Comment	Created: 1/5/2011...			
3	Set Context	WysiCorp Login/2.0			
4	Run Application				
6	Set Text	Guest	editboxUsername	WysiCorp...	
7	Key Press	KeyPress("Tab")			
8	Set Text	@*SjTy8o1+PaWEzo...	editboxPassword	WysiCorp...	
9	Click		buttonLogin	WysiCorp...	
10	Set Text	Guest	editbox1	WysiCorp...	
11	Key Press	KeyPress("Tab")			
12	Set Text	@*WhmcvKfER8bab7...	editbox2	WysiCorp...	
13	Click		Login	WysiCorp...	
14	Click		editboxFirst	WysiCorp...	
15	Click		buttonReset	WysiCorp...	

Associating datasheets from script properties

1. Select a script.
2. Choose **File > Properties**.

The Properties dialog box opens.



3. Select the datasheet from the **Data Source** list.
4. Click **OK** to save the changes.

Using datasheet values in scripts

After you associate a datasheet with a script, you can edit the script to use data from the datasheet. When a script runs, it uses the values from the first datasheet row and repeats the script until all rows are used.

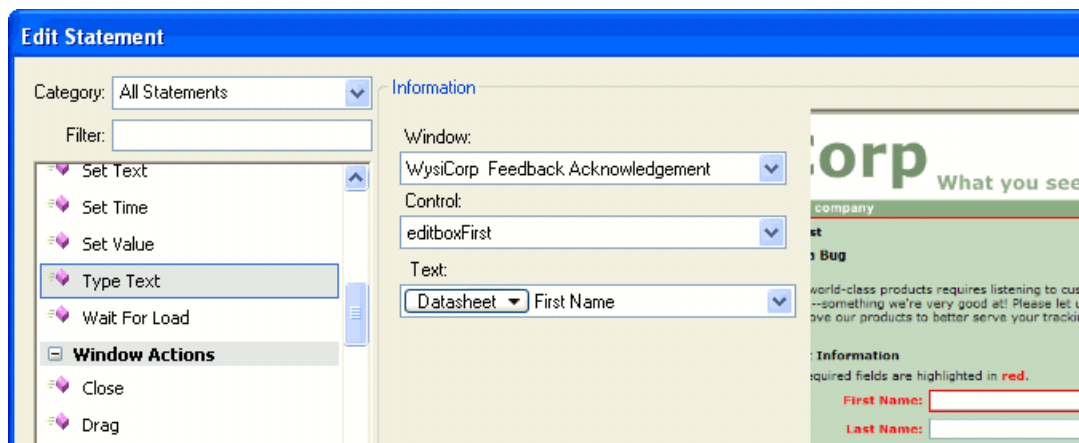
For example, a script tests a web site form. You create a local datasheet that contains a column for each field in the form and several rows. You associate the datasheet with the script and replace the data in the script with references to the corresponding columns in the datasheet. When the script runs, the value from the column in the current datasheet row is used to complete the corresponding field. The script repeats until all rows are used.

1. Select the step that contains the data you want to replace with a datasheet value.
2. Choose **Script > Edit Statement**.

The Edit Statement dialog box opens.

3. Select **Datasheet** in the field you want to use the datasheet value in and then select the appropriate column from the list.

Tip: To view the script and datasheet at the same time, right-click the script or datasheet tab in the Script pane and select New Horizontal Tab Group or New Vertical Tab Group.



4. Click **OK**.

The datasheet reference is added to the script step.

Using statements and functions in scripts

Datasheet statements and functions can be used in scripts to navigate, read from, and write to local datasheets, and limit the data in a datasheet.

Note: You cannot use datasheet statements and functions if the datasheet is associated with a script.

QA Wizard Pro includes the following statements for navigating and modifying datasheets.

Statement	Description
CommitRecordsetRow	Adds a data row to the recordset.
DeleteRecordsetRow	Deletes the current row from a recordset.
NextRow	Moves to the next row in a recordset.
PrevRow	Moves to the previous row in a recordset.
SetCell	Sets a column value in the current recordset or data row.
SetRowValue	Sets the value of a field in the recordset or data row.

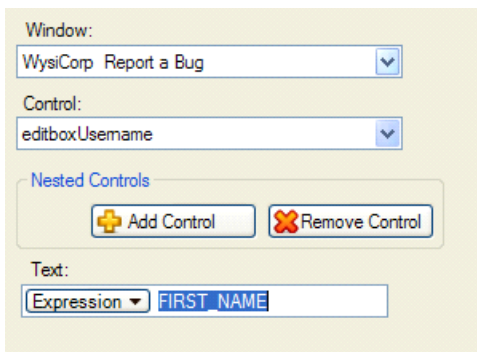
QA Wizard Pro includes the following functions for returning values from datasheets.

Function	Description
CreateRecordsetRow	Returns a new data row that can be populated and added to a recordset.
GetRowValue	Returns the column value in the current row in a recordset.
GetSQLScalar	Returns a value from a data source linked to an external datasheet based on a SQL scalar query.
OpenRecordset	Returns a recordset opened from a datasheet.
OpenRecordsetQuery	Returns a recordset from a data source linked to an external datasheet based on a SQL query.
RecordsetBOF	Returns true or false to indicate if the current recordset row index is at or before the beginning of the data.
RecordsetEOF	Returns true or false to indicate if the current recordset row index is past the end of the data.
RecordsetRowCount	Returns the total number of records in a recordset.
RunSQLCommand	Runs a SQL command on a data source linked to an external datasheet and returns the number of affected rows.

1. Add the statements to perform to the script.
2. Before the statements you want to use datasheet values in, add a variable equal to the OpenRecordset function and datasheet to access. The OpenRecordset function syntax is `OpenRecordset ("DatasheetName")`.
For example, to add the variable for a local datasheet named CustomerInfo, enter `Recordset = OpenRecordset ("CustomerInfo")`.

Note: If the datasheet uses a script-generated query to retrieve data during playback, add a variable equal to the OpenRecordsetQuery function.

3. Add variables for each datasheet column used in the script. Set each variable equal to the GetRowValue function. The GetRowValue function syntax is `GetRowValue (Recordset, "ColumnName")`.
For example, to add the variable for the first name field, enter `FIRST_NAME = (GetRowValue (Recordset, "First Name"))`.
4. Choose **Script > Edit Statement**.
The Edit Statement dialog box opens.
5. Select **Expression**.
6. Enter the variable that contains the datasheet column to use.



7. Click **OK**. The datasheet reference is added to the statement.

Editing local datasheets

You can edit your local datasheets to change the column names and descriptions or add additional columns to the datasheet.

1. Select the datasheet and choose **Data > Edit Datasheet**.
2. The Edit Datasheet dialog box opens.
3. Make any changes.

Tip: Right-click and choose Insert Row or Delete Row to add or remove a column.

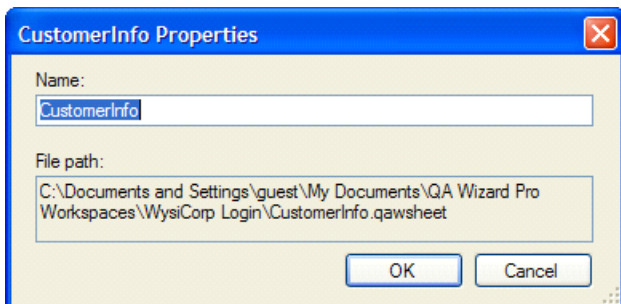
4. Click **OK**.
The datasheet is updated.

Modifying local datasheet properties

You can rename a local datasheet or view the location where it is saved.

1. Select the local datasheet in the Workspace pane.
2. Choose **File > Properties**.

The Properties dialog box opens.



3. Optionally enter a new datasheet **Name**.
If you rename the datasheet, it is no longer associated with scripts. You must associate the renamed datasheet with your scripts.
4. Click **OK** to save the changes.

Local Datasheets Conclusions

Using local datasheets allows you to save time by testing a wide variety of data using one or more testing scripts. Using a combination of scripts and local datasheets, you can thoroughly test your application and provide a more stable product to your users.

Links to other resources

You can view the following resources for more information on QA Wizard Pro.

- [QA Wizard Pro Resource Center](http://www.seapine.com/qawealtools.php) - <http://www.seapine.com/qawealtools.php>
- [QA Wizard Pro Blog](http://blogs.seapine.com/category/products/qawizardpro/) - <http://blogs.seapine.com/category/products/qawizardpro/>
- [Knowledgebase](http://www.seapine.com/kb/categories/QA+Wizard+Pro/) - <http://www.seapine.com/kb/categories/QA+Wizard+Pro/>