

## External Datasheets How To

You can use QA Wizard Pro to test multiple scenarios with large amounts of data by linking to external test data stored in Microsoft Access, Microsoft Excel, Microsoft SQL Server, MySQL, Oracle, or text files. This How To walks you through creating an external datasheet, associating it with a script or using datasheet statements and functions, and running the script using the datasheet values.

## External Datasheets Overview

You can test a large amount of data with one script by creating an external datasheet and linking to it in QA Wizard Pro or by importing an existing external datasheet. After the datasheet is linked with a script or imported, 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 external 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 linking an external datasheet with a script, you can:

- Complete tests with a large amount of data using one script
- Use existing test data instead of re-creating it
- Easily modify test data and the scripts that use it
- Maintain test data outside of QA Wizard Pro
- Test applications with dynamic user interface elements

## External Datasheets Scenario

This external datasheets scenario describes how to create an external datasheet and link to it from QA Wizard Pro, import an existing external datasheet, assign datasheet values in your scripts, use datasheet statements and functions, and modify external datasheet properties.

## Setup

Before you begin this external 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.
- **Set up external data source access**—Make sure you have access to the external data sources you need to access from QA Wizard Pro.

## External Datasheets Instructions

QA Wizard Pro can retrieve test data from external sources by importing data from an external source or by linking to the external data.

You can retrieve test data from the following sources:

- Microsoft Access—See [Creating Microsoft Access datasheets, page 2](#).
- Microsoft Excel—See [Creating Microsoft Excel datasheets, page 5](#).
- Microsoft SQL Server—See [Creating Microsoft SQL Server datasheets, page 6](#).
- MySQL—See [Creating MySQL datasheets, page 9](#).
- Oracle—See [Creating Oracle datasheets, page 11](#).
- Text files—See [Creating text file datasheets, page 13](#).

## Configuring external datasheets

If you want to copy test data into a datasheet that can be changed in QA Wizard Pro, you can import test data from an external source. This option is useful if you are creating new scripts because you can run the scripts without modifying the data in the external source. After the data is imported into QA Wizard Pro, you can modify the data locally to test different scenarios.

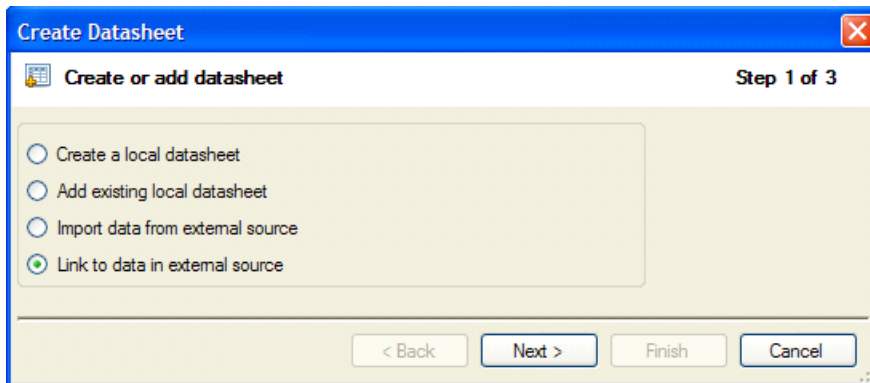
If you want to modify the test data only in the external data source, you can link to the data. When changes are made to the external data source, the data in the linked datasheet is automatically updated. This option is useful for updating test data without interrupting an established testing process.

## Creating Microsoft Access datasheets

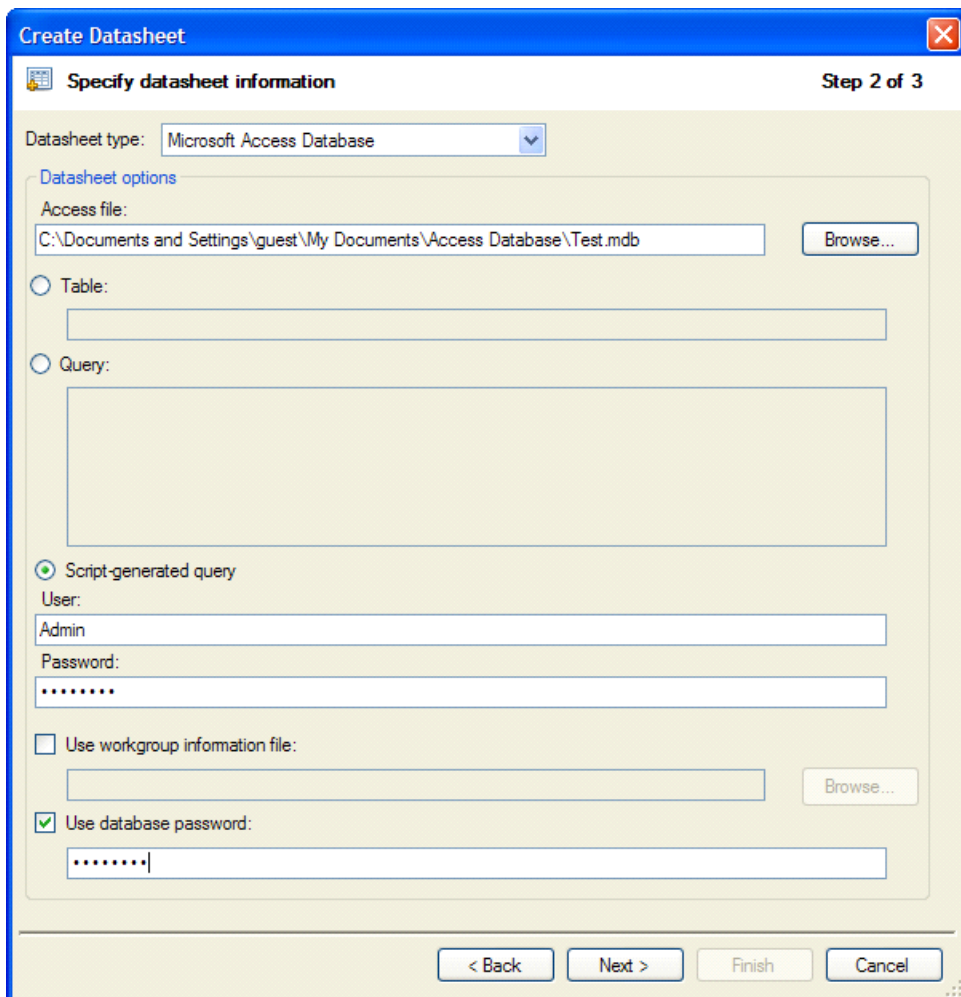
QA Wizard Pro can connect to a Microsoft Access database and retrieve data to use during testing. You can retrieve all data from a table, use a query to retrieve a subset of data, or use a script-generated query to retrieve data during playback.

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

The Create Datasheet wizard opens.



2. Specify the data source.
  - Select **Import data from external source** to import the Access data into QA Wizard Pro. Select this option if you want to change the test data in QA Wizard Pro without affecting the Access database.
  - Select **Link to data in external source** to link to the Access data. Select this option if you want to maintain the test data in Access.
3. Click **Next**.
4. Select **Microsoft Access Database** from the Datasheet type list.
5. Enter the path to the Access file or click **Browse** to select the file.



6. Specify the data you want to retrieve from the database.
  - Select **Table** to retrieve all data from a specified table.
  - Select **Query** to retrieve a subset of data from a table. The query must return data from the source database. You can create the query in Access, copy it, and paste it into the Query field.
  - Select **Script-generated query** to use a function in a script to retrieve a subset of data during playback. This option is only available if the datasheet links to an external data source. If this option is selected, you cannot open the datasheet for viewing in QA Wizard Pro.
7. Enter the **User** and **Password** if user-level security is configured for the database. The default Access user is Admin with no password.
8. Select **Use workgroup information file** if the database uses a workgroup information file for security. Click **Browse** to select the file.
9. Select **Use database password** if users must enter a password to access the database. Enter the password.
10. Click **Next**.
11. Specify the datasheet properties.

- If you are importing the data, select a **Datasheet Path** and enter a datasheet name. You can also optionally enter a datasheet **Description**.
- If you are linking to the Access database, enter a **Datasheet Name**.

12. Click **Finish**.

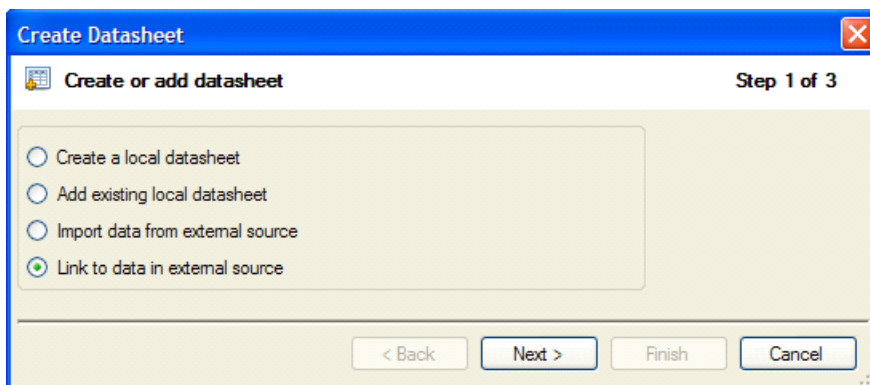
The datasheet is created and added to your Workspace.

## Creating Microsoft Excel datasheets

QA Wizard Pro can connect to a Microsoft Excel sheet and retrieve data to use during testing. You can retrieve all data from an Excel sheet or use a script-generated query to retrieve data during playback.

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

The Create Datasheet wizard opens.



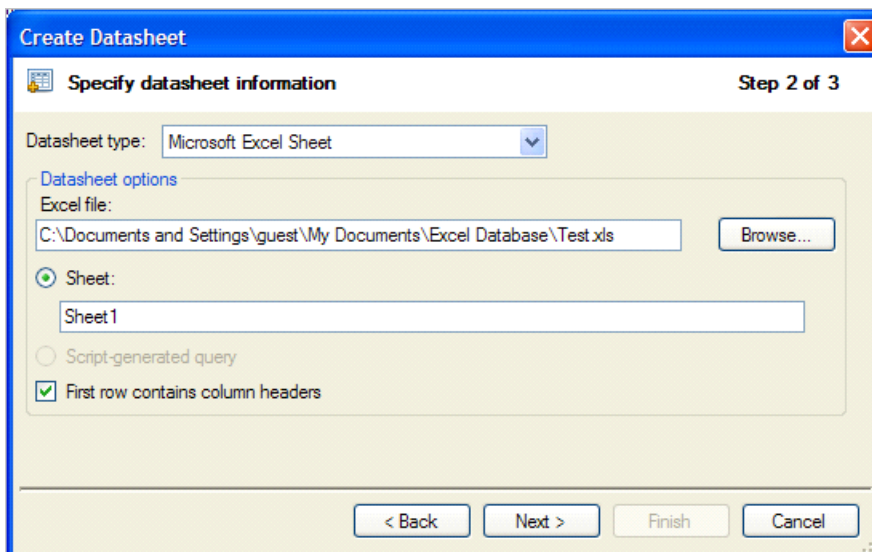
2. Specify the data source.

- Select **Import data from external source** to import the Excel data into QA Wizard Pro. Select this option if you to change the test data in QA Wizard Pro without affecting the Excel database.
- Select **Link to data in external source** to link to the Excel data. Select this option if you to maintain the test data in Excel.

3. Click **Next**.

4. Select **Microsoft Excel Sheet** from the Datasheet type list.

5. Enter the path to the Excel file or click **Browse** to select the file.

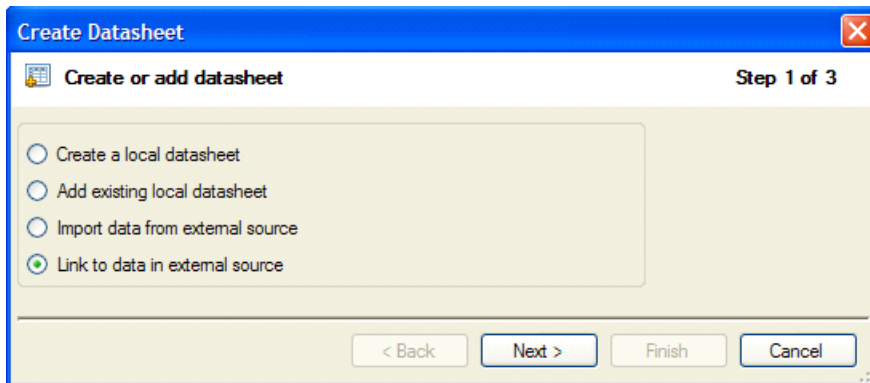


6. Specify the data you want to retrieve from the spreadsheet.
  - Select **Sheet** to retrieve all data from a specified worksheet in the spreadsheet.
  - Select **Script-generated query** to use a function in a script to retrieve a subset of data during playback. This option is only available if the datasheet links to an external data source. If this option is selected, you cannot open the datasheet for viewing in QA Wizard Pro.
7. Select **First row contains column headers** if the Excel spreadsheet contains column headers. If this option is not selected, column names are created when the datasheet is added.
8. Specify the datasheet properties.
  - If you are importing the data, select a **Datasheet Path** and enter a datasheet name. You can also optionally enter a datasheet **Description**.
  - If you are linking to the Excel spreadsheet, enter a **Datasheet Name**.
9. Click **Finish**.  
The datasheet is created and added to your Workspace.

## Creating Microsoft SQL Server datasheets

QA Wizard Pro can connect to a Microsoft SQL Server database and retrieve data to use during testing. You can retrieve all data from a table, use a query to retrieve a subset of data, or use a script-generated query to retrieve data during playback.

1. Choose **Data > Create Datasheet**.  
The Create Datasheet wizard opens.



2. Specify the data source.
  - Select **Import data from external source** to import the SQL data into QA Wizard Pro. Select this option if you want to change the test data in QA Wizard Pro without affecting the SQL database.
  - Select **Link to data in external source** to link to the SQL data. Select this option if you want to maintain the test data in SQL.
3. Click **Next**.
4. Select **Microsoft SQL Server Database** from the Datasheet type list.
5. Enter the **Server** and **Database** names.

The screenshot shows a 'Create Datasheet' dialog box with the following fields and options:

- Datasheet type:** Microsoft SQL Server Database
- Datasheet options:**
  - Server:** 192.168.1.100
  - Database:** Testing Database
  - Table:** Test Table (selected with a radio button)
  - Query:** (empty text area, unselected)
  - Script-generated query:** (unselected)
- Authentication type:** Use SQL Server Authentication
- User:** Test User
- Password:** (masked with dots)
- Network protocol:** TCP/IP
- TCP/IP port:** 1433

Buttons at the bottom: < Back, Next >, Finish, Cancel.

6. Specify the data you want to retrieve from the database.
  - Select **Table** to retrieve all data from a specified table.
  - Select **Query** to retrieve a subset of data from a table. The query must return data from the source database. You can create the query in SQL Server, copy it, and paste it in the Query field.
  - Select **Script-generated query** to use a function in a script to retrieve a subset of data during playback. This option is only available if the datasheet links to an external data source. If this option is selected, you cannot open the datasheet for viewing in QA Wizard Pro.
7. Select an **Authentication Type**.
  - Select **Use Windows Authentication** to use the username and password of the logged in user to validate the SQL Server account name and password.
  - Select **Use SQL Authentication** to use the username and password saved on the SQL Server. Enter the SQL Server **User** and **Password**.
8. Select a **Network Protocol**.

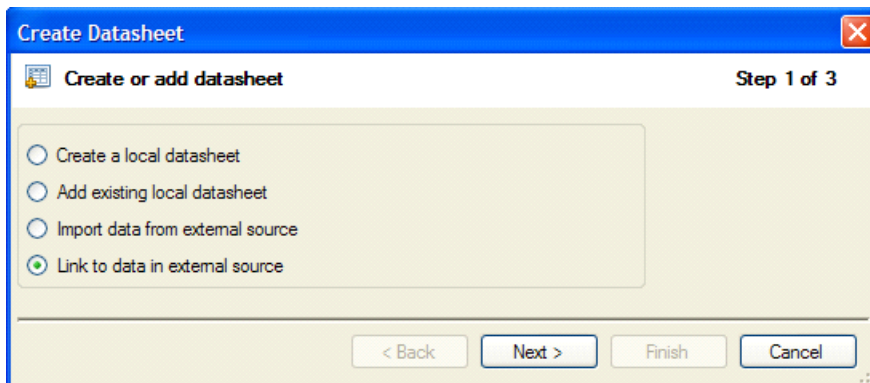
- Select **Named pipes** if the SQL Server uses pipes to communicate.
  - Select **TCP/IP port** if the SQL Server uses TCP/IP to communicate. Enter the **TCP/IP port**.
9. Specify the datasheet properties.
- If you are importing the data, select a **Datasheet Path** and enter a datasheet name. You can also optionally enter a datasheet **Description**.
  - If you are linking to the SQL Server database, enter a **Datasheet Name**.
10. Click **Finish**.
- The datasheet is created and added to your Workspace.

## Creating MySQL datasheets

QA Wizard Pro can connect to a MySQL database and retrieve data to use during testing. You can retrieve all data from a table, use a query to retrieve a subset of data, or use a script-generated query to retrieve data during playback.

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

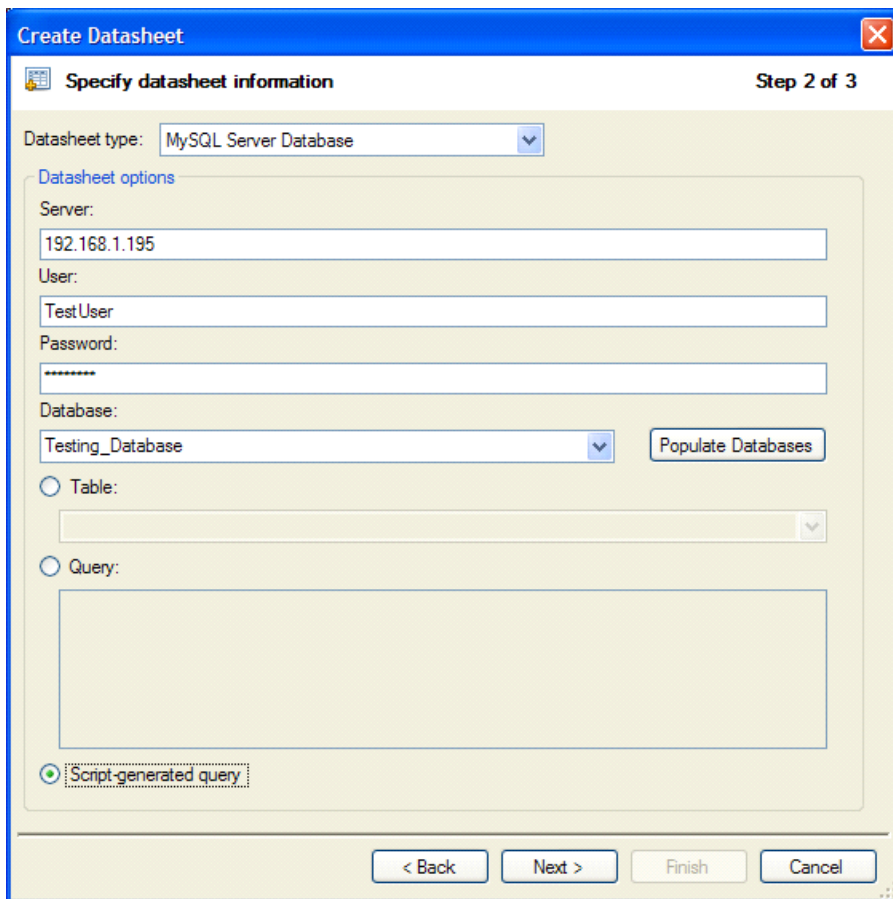
The Create Datasheet wizard opens.



2. Specify the data source.
  - Select **Import data from external source** to import the MySQL data into QA Wizard Pro. Select this option if you want to change the test data in QA Wizard Pro without affecting the MySQL database.
  - Select **Link to data in external source** to link to the MySQL data. Select this option if you want to maintain the test data in MySQL.
3. Click **Next**.
4. Select **MySQL Server Database** from the Datasheet type list.

**Note:** You are prompted to download the MySQL Connector/Net (ADO.NET driver for MySQL) if it is not installed. The driver can be downloaded from [www.mysql.org/downloads/connector/net/](http://www.mysql.org/downloads/connector/net/).

5. Enter the **Server Name**.



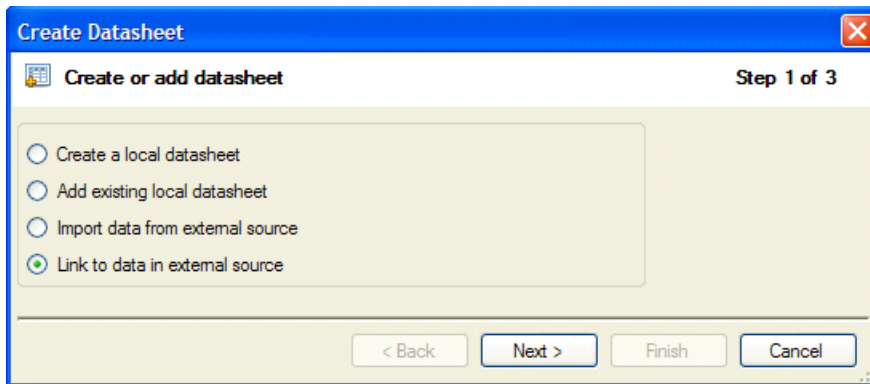
6. Enter the MySQL server **User** and **Password** if security is configured for the database.
7. Enter the **Database** name. Click **Populate Databases** to retrieve a list of databases from the MySQL server.
8. Specify the data you want to retrieve from the database.
  - Select **Table** to retrieve all the data from a specified table.
  - Select **Query** to retrieve a subset of data from a table. The query must return data from the source database. You can create the query in MySQL, copy it, and paste it in the Query field.
  - Select **Script-generated query** to use a function in a script to retrieve a subset of data during playback. This option is only available if the datasheet links to an external data source. If this option is selected, you cannot open the datasheet for viewing in QA Wizard Pro.
9. Specify the datasheet properties.
  - If you are importing the data, select a **Datasheet Path** and enter a datasheet name. You can also optionally enter a datasheet **Description**.
  - If you are linking to the MySQL database, enter a **Datasheet Name**.
10. Click **Finish**.  
The datasheet is created and added to your Workspace.

## Creating Oracle datasheets

QA Wizard Pro can connect to an Oracle database and retrieve data to use during testing. You can retrieve data from a table, use a query to retrieve a subset of data, or use a script-generated query to retrieve data during playback. QA Wizard Pro supports Oracle 8i release 3 and later databases.

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

The Create Datasheet wizard opens.



2. Specify the data source.
  - Select **Import data from external source** to import the Oracle data into QA Wizard Pro. Select this option if you want to change the test data in QA Wizard Pro without affecting the Oracle database.
  - Select **Link to data in external source** to link to the Oracle data. Select this option if you want to maintain the test data in Oracle.
3. Click **Next**.
4. Select **Oracle Database** from the Datasheet type list.

The screenshot shows a dialog box titled "Create Datasheet" with a close button in the top right corner. The dialog is divided into sections for specifying connection and authentication details.

**Specify datasheet information** (Step 2 of 3)

Datasheet type: Oracle Database

**Datasheet options**

- TNS alias connection  
TNS alias: [Empty text box]
- Basic connection  
Server address: 192.168.5.55  
Server port: 1521  
Connect via Service Name: [Dropdown menu showing "Connect via Service Name"]  
SID / Service name: Test\_2011

**Authentication type:** Use specified user and password

User: Test User  
Password: [Masked with dots]

**Source**

- Table: [Empty text box]
- Query: [Empty text area]
- Script-generated query

Navigation buttons: < Back, Next >, Finish, Cancel

5. Select a database connection type and enter the connection details.
  - Select **TNS alias connection** to use a TNS alias to connect to the database.
  - Select **Basic connection** to specify the **Server address** and **Server port**. The default Oracle port is 1521. Select a database connection option and enter the **SID/Service name**. **Connect via SID** connects to the database using a system identifier. **Connect via Service Name** connects to the database using the service name.
6. Select an **Authentication type**.

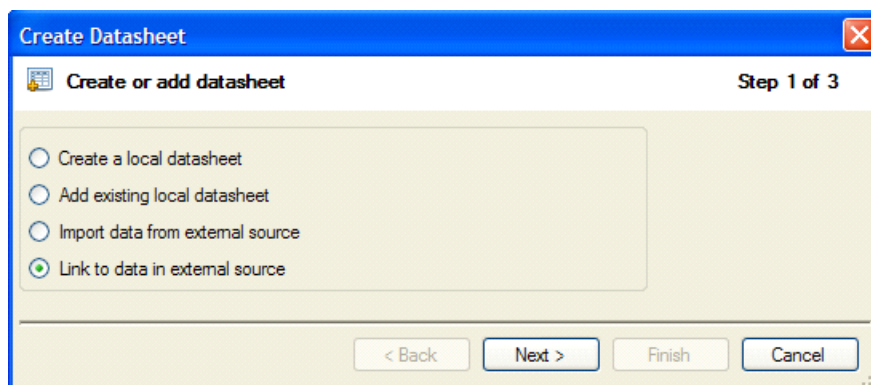
- Select **Use integrated security** to use built-in Oracle authentication to access the database.
  - Select **Use specified user and password** to use a specific username and password to access the database. Enter the **User** and **Password**.
7. Specify the data you want to retrieve from the database.
    - Select **Table** to retrieve all data from a specified table.
    - Select **Query** to retrieve a subset of data from a table. The query must return data from the source database. You can create the query in Oracle, copy it, and paste it in the Query field.
    - Select **Script-generated query** to use a function in a script to retrieve a subset of data during playback. This option is only available if the datasheet links to an external data source. If this option is selected, you cannot open the datasheet for viewing in QA Wizard Pro.
  8. Specify the datasheet properties.
    - If you are importing the data, select a **Datasheet Path** and enter a datasheet name. You can also optionally enter a datasheet **Description**.
    - If you are linking to the Oracle database, enter a **Datasheet Name**.
  9. Click **Finish**.
- The datasheet is created and added to your Workspace.

## Creating text file datasheets

QA Wizard Pro can connect to a text file and retrieve data to use during testing. QA Wizard Pro supports comma-delimited and fixed-width text files.

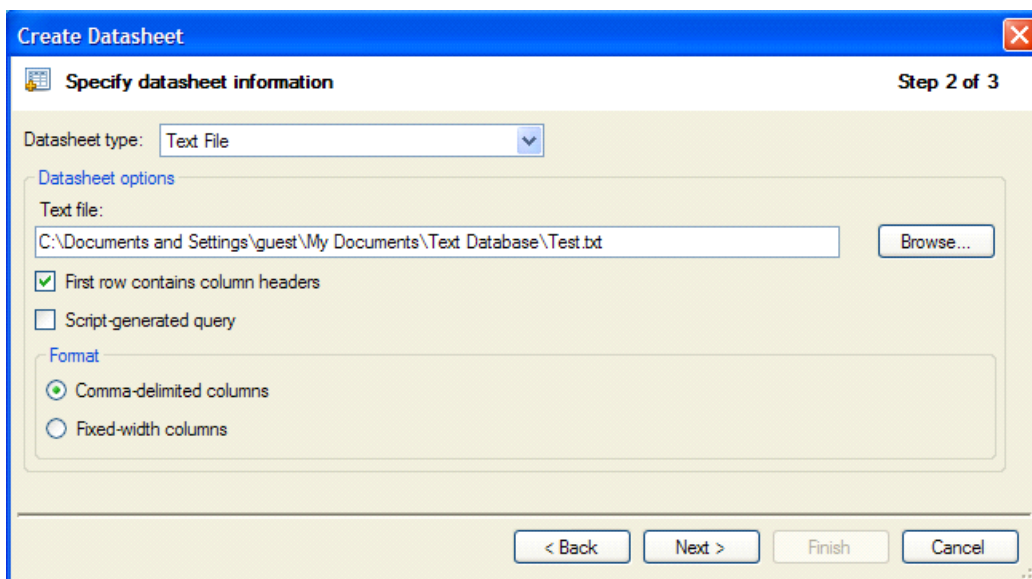
1. Choose **Data > Create Datasheet**.

The Create Datasheet wizard opens.



2. Specify the data source.
  - Select **Import data from external source** to import the data into QA Wizard Pro. Select this option if you want to change the test data in QA Wizard Pro without affecting the text file.
  - Select **Link to data in external source** to link to the data. Select this option if you want to maintain the test data in the text file.
3. Click **Next**.
4. Select **Text File** from the Datasheet type list.

5. Enter the path to the text file.



6. Select **First row contains column headers** if the text file contains column headings. If this option is selected, QA Wizard Pro uses the values in the first row of the file to identify columns. If it is not selected, column names are created when the datasheet is added.
7. Select the text file format.
  - Select **Comma-delimited columns** if the columns are separated by commas.
  - Select **Fixed-width columns** if the columns are set at a fixed width. In some instances, you need to create a schema.ini file to collect all the data from a text file. If you are using a comma-delimited text file that contains different data types in the same column, you must create a schema.ini file to identify the types of data in the file. If you are using a fixed-width text file, you must always create a schema.ini file to identify the file contents and column width. This file must be saved in the same directory as the text file.
8. Select **Script-generated query** to use a function in a script to retrieve a subset of data from the text file during playback. This option is only available if the datasheet links to an external data source. If this option is selected, you cannot open the datasheet for viewing in QA Wizard Pro.
9. Specify the datasheet properties.
  - If you are importing the data, select a **Datasheet Path** and enter a datasheet name. You can also optionally enter a datasheet **Description**.
  - If you are linking to the text file, enter a **Datasheet Name**.
10. Click **Finish**.

The datasheet is created and added to your Workspace.

### Create a schema.ini file for text files

To use a fixed-width text file or a comma-delimited text file that contains mixed data types in the same column as a data source, you need to create a schema.ini file to identify the contents of the file.

The schema file includes entries that specify characteristics of the text file including the file name, file format, field names, widths, and types, character set, and data type conversions. You can use any text editor to create the schema file, which must be stored in the same directory as the text file.

The following example shows the information that must be included in the schema file for a fixed-width text file.

```
[filename.txt]
Format=FixedLength
ColNameHeader=False
Col1=ColName1 TEXT Width 15
Col2=ColName2 TEXT Width 15
Col3=ColName3 TEXT Width 40
Col4=ColName4 TEXT Width 20
CharacterSet=ANSI
```

The following example shows the information that must be included in the schema file for a comma-delimited text file that contains mixed data types.

```
[filename.txt]
Format=CSVDelimited
ColNameHeader=False
Col1=ColName1 TEXT
Col2=ColName2 TEXT
CharacterSet=ANSI
```

Item	Description
Filename	Text file name; must be enclosed in brackets
Format	Text file format: FixedLength or CSVDelimited
ColNameHeader	Indicates if the first record in the text file contains column headings (True) or not (False)
Col	Column by number; required for fixed-width files but optional for comma-delimited files
ColName	Column name; used to identify the column in QA Wizard Pro
DataType	Type of data in the column: <ul style="list-style-type: none"> <li>■ TEXT - Text values; unspecified length returns 255 bit width</li> <li>■ FLOAT - Signed, approximate, numeric values; maximum width includes the sign and decimal point</li> <li>■ INTEGER - Non-decimal numeric values (whole numbers); valid width values are 32767 to -32766</li> </ul>

Item	Description
	<ul style="list-style-type: none"> <li>■ LONGCHAR - Text values; unspecified length returns 65500 KB width</li> <li>■ DATETIME - String that specifies date/time format</li> </ul>
Width	Column width
CharacterSet	Character set: ANSI or OEM

## Using external datasheets in 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

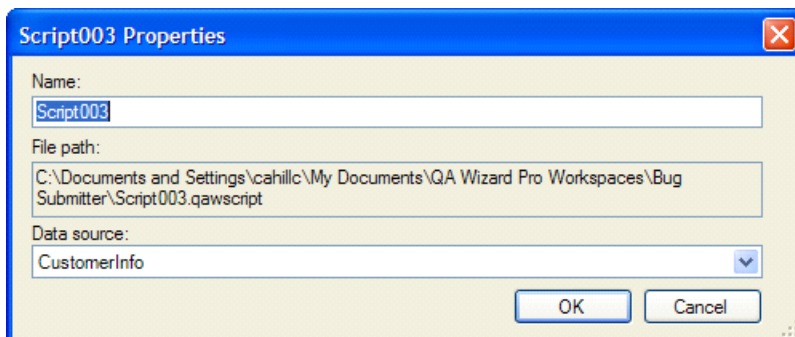
1. Open the script you want to associate the datasheet with.
2. Select the datasheet from the **Data Source** list in the Script pane.

#	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 have already created an external datasheet that contains a column for each field in the form and several rows. You link to the datasheet from QA Wizard Pro 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.

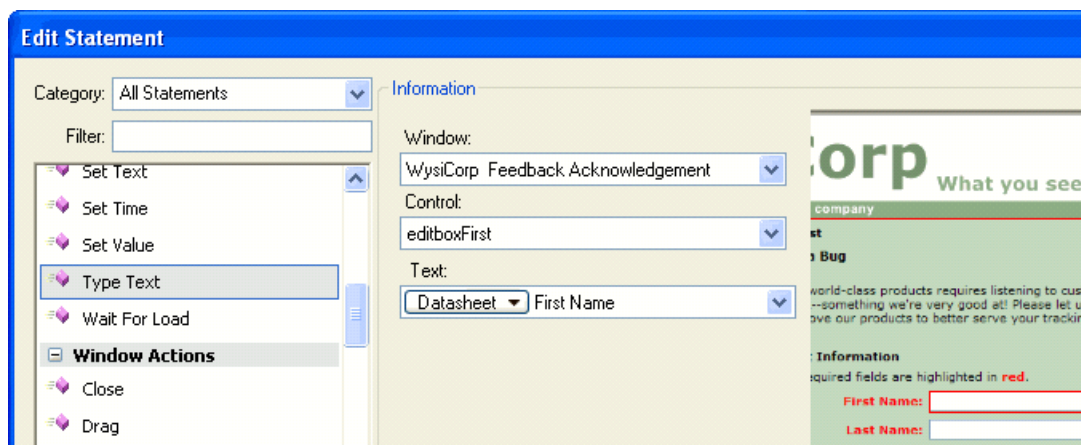
1. Select the step that contains the data you want to replace with a datasheet value.

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 datasheet statements and functions in scripts

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

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

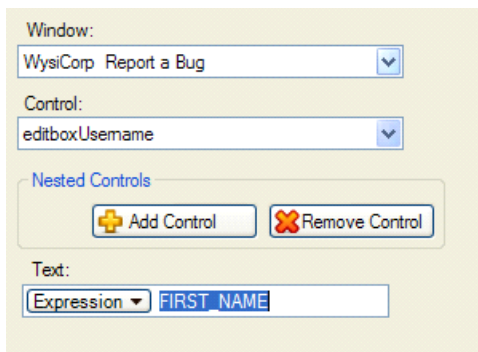
QA Wizard Pro includes the following statements for navigating and modifying datasheets.

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 an external datasheet named CustomerInfo, enter `Recordset = OpenRecordset ("CustomerInfo")`.
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.

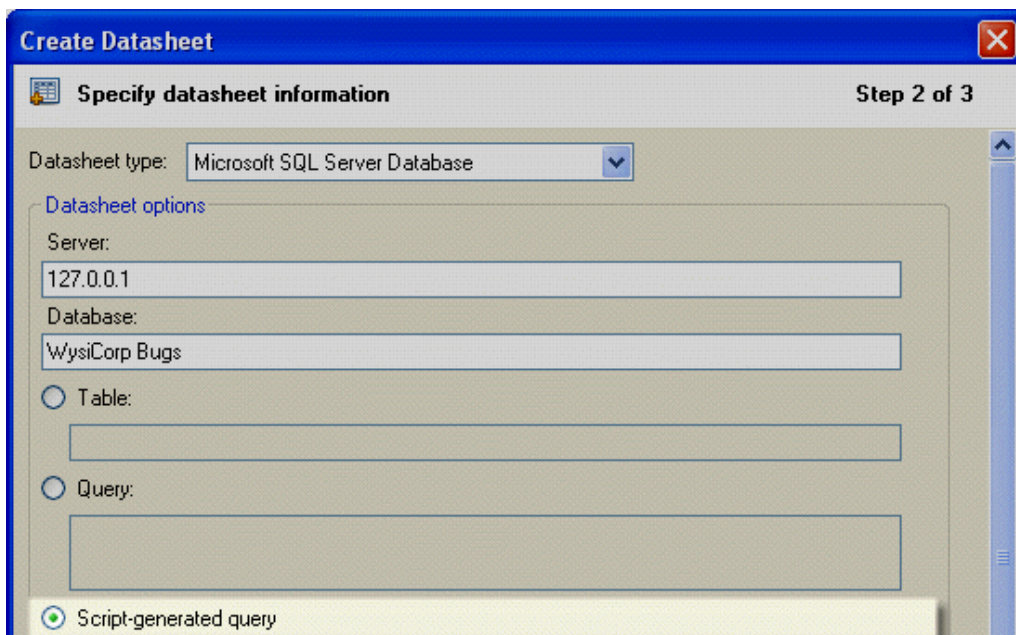
### Using script-generated database queries

If you use multiple sets of data from one external data source, you can configure a single external datasheet and use a script-generated query to retrieve a recordset during playback instead of creating multiple external datasheets to retrieve multiple recordsets. This allows you to perform a dynamic query on a data source during playback.

Script-generated queries can only be used for external datasheets linked to a data source.

You can use the OpenRecordsetQuery function to a script to perform the query on the data source.

1. Choose **Data > Create Datasheet**.
2. Select **Link to data in external source**.
3. Click **Next**.
4. Select **Script-generated query** in the datasheet information.



5. Click **Next**.
6. Enter a **Datasheet Name**.

7. Click **Finish**.

The datasheet is created and added to your Workspace.

8. Add the OpenRecordsetQuery function to a script to specify the data to open from the external data source during playback.

### Example

In the following script, the OpenRecordsetQuery function uses the SELECT \* FROM Email SQL query to retrieve all columns and rows from the Email table in the database linked to the Customers external datasheet. The recordset is assigned to the RECORDSET variable. The RecordsetRowCount function returns the number of rows in the recordset and assigns the number to the COUNT variable. The PrintLn statement prints the number of rows stored in the COUNT variable.

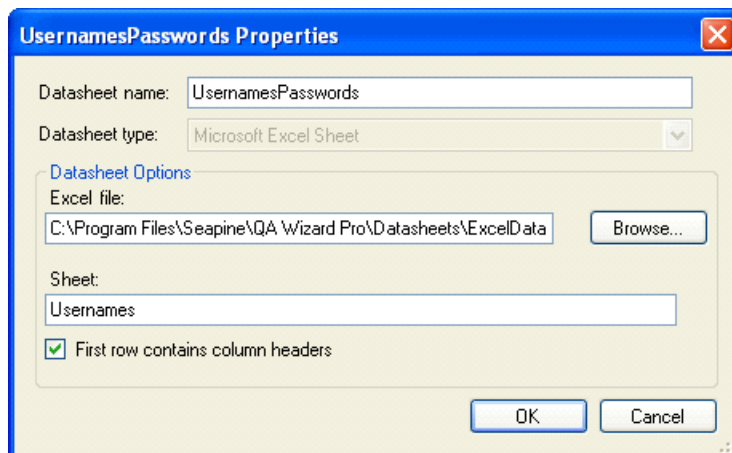
```
RECORDSET = OpenRecordsetQuery("Customers", "SELECT * FROM Email")
COUNT = RecordsetRowCount(RECORDSET)
PrintLn(COUNT)
```

## Modifying external datasheet properties

You can rename an external datasheet, change its data source, or select a different set of data.

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

The Properties dialog box opens. The fields are different depending on the datasheet type.



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 scripts.

4. Make any changes to the **Datasheet Options**.

You cannot change the datasheet type.

5. Click **OK** to save the changes.

## External Datasheets Conclusions

Using external datasheets allows you to save time by testing a wide variety of data using one or more testing scripts. Using a combination of scripts linked to external 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/qawevaltools.php) - <http://www.seapine.com/qawevaltools.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/>