Datasets and Data Exchange

This page explains how to setup your database in FactoryStudio, using the built-in internal database or connecting to an existing external database.

External databases can be configured on the same server that is running the project or on different servers, using remote configuration.

Different providers can be used (SQL Server, Oracle, SQLite, PostgreSQL, and others...), and can be simply and quickly configured.

Creating a Connection

By default, FactoryStudio uses an embedded SQL database (TatsoftDB) for the tag and alarm historians. You can configure external databases such as Microsoft SQL Server, Oracle, or others. If you expect your database to grow to more than 10 GB, you should consider using an external database.

When using the embedded database, the system automatically creates the database file. When using external databases, the database must already exist. However, FactoryStudio can create the tables in the database.

You can also use external databases as a data source in your application to display such things as customer or product information or to configure systems from recipes.

You must be logged in as an Administrator in FactoryStudio to configure the database login and password.

To configure an external database:

- Go to **Edit >Datasets** > **DBs**.
- Click **Create New**. The Create New Database Connection window displays.
- Enter or select information, as needed.

- Click **OK**. The database is added as a new row in the table.
- Enter or select information, as needed.

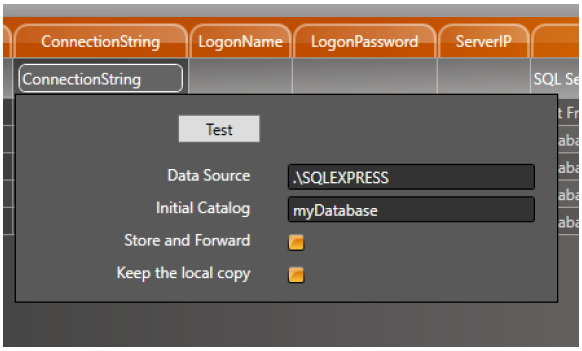
Column	Description
Name	Enter a name for the database configuration. The system lets you know if the name is not valid.
Provider	Select the database provider.
Database	If options display here, select the database type.
ConnectionString	Enter the information needed to connect to the database.
LogonName	Enter a valid login name for the database.
LogonPassword	Enter the password that corresponds to the database login.
ServerIP	Enter the database's server IP.
Description	Enter a description for the database.
[Other columns]	For definitions of other columns that are available in many tables, see Common Column Definitions.

- If you are using a new external database for the alarm or tag historian, rename the existing TagHistorian or AlarmHistorian database to something else. Then rename the new database as TagHistorian or AlarmHistorian. For the tag or alarm historian databases, the database configuration on the tab must be named TagHistorian or AlarmHistorian. You cannot have two databases with the same name.
- Continue adding as many database configurations as you need.

The databases you create are available to be used in displays and scripts. You can find them on the **Tags > Historian** tab.

- If needed, right-click a row to cut, copy, paste, or delete the row.

Historian and Alarms Connection String



The connection string for the TagHistorian and AlarmHistorian has parameters that can be configured to avoid data loss. They are:

- Data Source: The server path and instance that will have the databases.
- Initial Catalog: The name of the database that will be used.
- Store and Forward: Enabling this option will cause the system to store the data locally if communication with the database is lost, and forwards the data to synchronize once the connection is back again.
- Keep the local copy: Enabling this option will make the system store a local database with the same data that has been stored in the main DB. The system will use the internal database to store it, and the file can be located on the same folder that the project is in.

Accessing Microsoft Excel

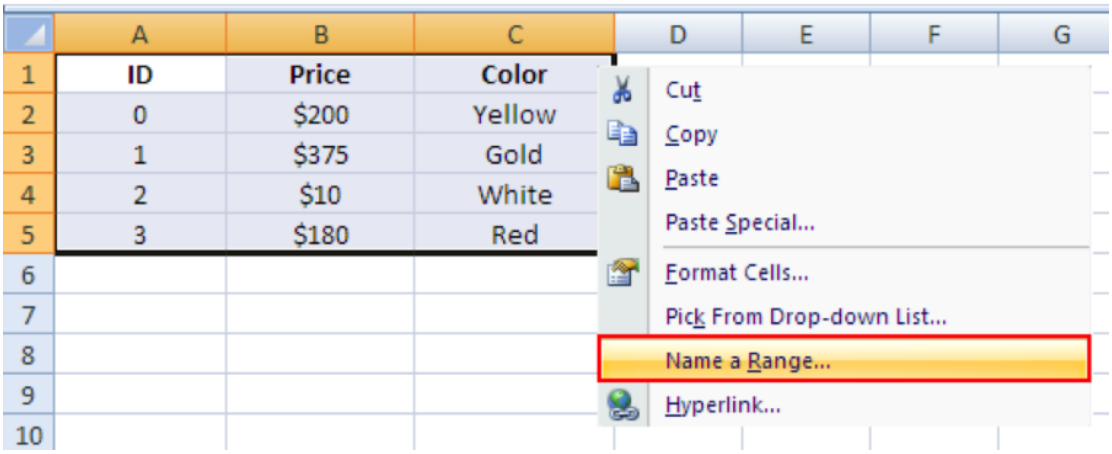
You can connect to Excel databases using an ODBC driver, an ODBC DSN, or OleDB.

Using ODBC

Select and name a range of rows and columns in the worksheet. This will allow the software to read the information as a table. Choose one of the following naming processes for your version of Microsoft Excel:

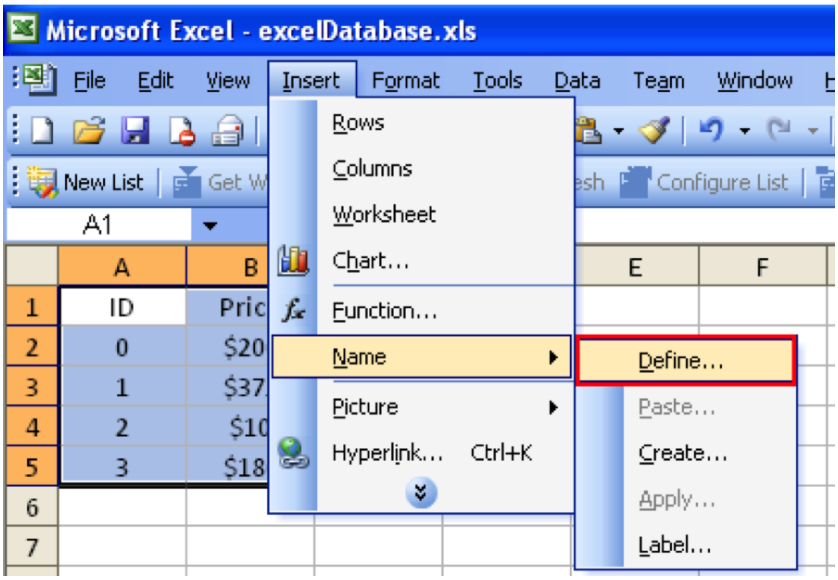
For Microsoft Office 2007:

Right-click the selection and choose "Name a Range".



For Microsoft Office 2003:

In Microsoft Excel, go to "Insert > Name > Define".

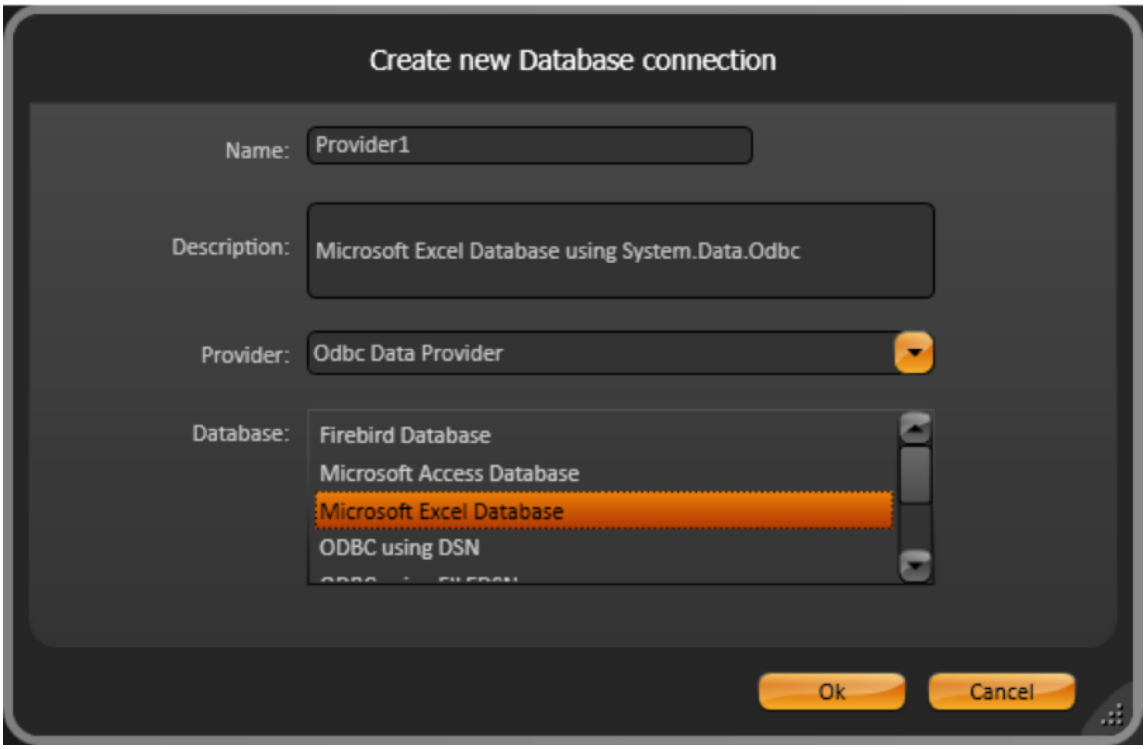


Name the selection (e.g., "itemsTable"). The Excel file is ready to be used.

There are several methods for communicating using ODBC:

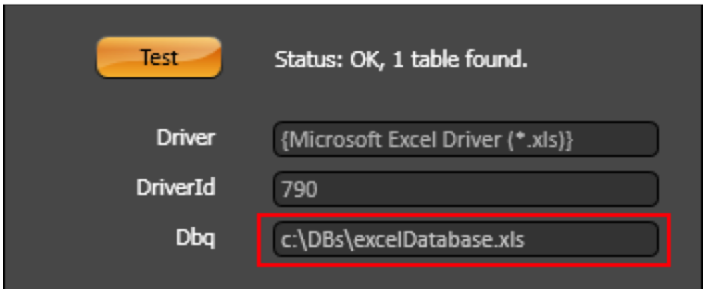
Using ODBC Microsoft Excel Driver

- In the datasets namespace, choose the "DBs" tab and create a new provider by clicking Create new.
- Select the "Odbc Data Provider" in the "Provider" data field.
- For the "Database" field, choose "Microsoft Excel Database".
- Click Ok.



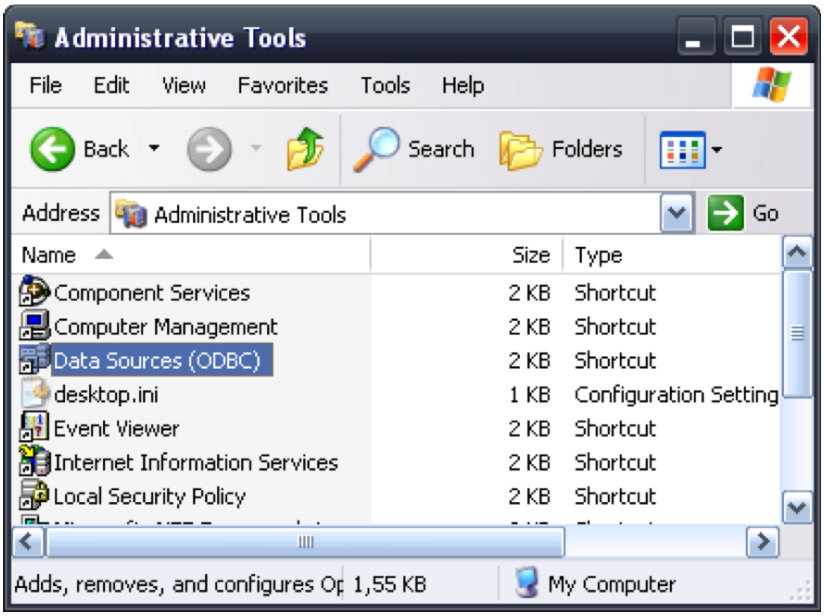
- A new row is created in the data grid, click the "ConnectionString" column.
- In the pop-up window, enter the path and the filename in the "Dbq" field.
- Click the "Test" button to ensure that the connection is OK.

Note: "Test" is optional.

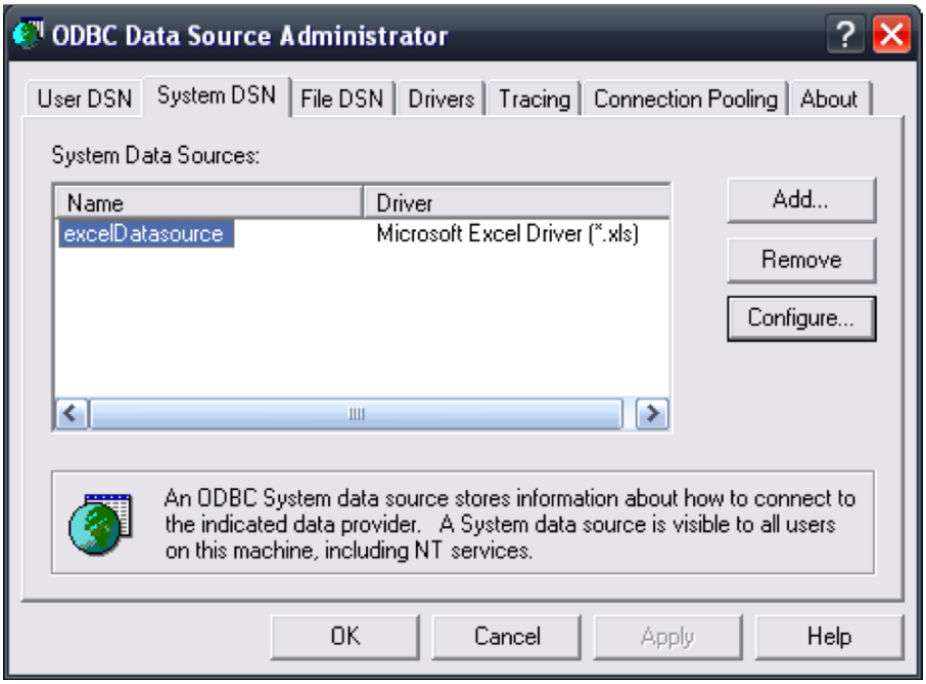


Using ODBC with a DSN

- Go to your computer's "Control Panel" and select "Administrative Tools".
- Double-click on "Data Sources (ODBC)".



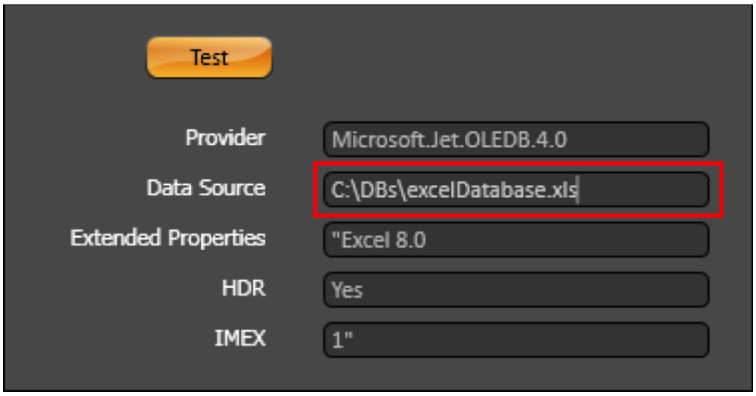
- In the "ODBC Data Source Administrator" window, click Add. You are prompted to select a driver.
- Select the "Microsoft Excel Driver (*.xls)".
- Click Select Workbook and select the name of the Excel file that was previously created.
- Name the Data Source, (e.g., "excelDatasource").
- For Write access, uncheck the "ReadOnly" checkbox.



- In the Datasets namespace, choose the "DBs" tab and create a new provider by clicking create new.
- Under the "Odbc Data Provider" options, choose "ODBC using DSN". Then, click Ok.
- Click the ConnectionString column of the new row. Then, enter the DSN in the "DSN" field.

Using OleDB

- In the Datasets namespace, choose the "DBs" tab.
- Select the "OleDb data provider" option in the combo-box and create a new provider by clicking create new.
- Choose the "Microsoft Excel Database" . Then, click Ok.
- Click the ConnectionString column of the new row, then enter the path and the name of the Excel (.xls) file in the "DataSource" field.



Working with Database Tables

If you are using an external database as a data source in your application, you can specify which table you want to use from the database.

To configure database tables:

- Go to **Edit >Datasets** > **Tables**.
- Enter or select information, as needed.

Column	Description
Name	Enter a name for the table configuration. The system lets you know if the name is not valid.
DB	Select the database configuration.
TableName	Select the table name.
WhereCondition	Specify the parameters that will filter the data using SQL syntax. E.g. "ColumnName = {@tag.tagInt}"
Access	Select the access permissions for the table.
Mapping	Click ... to select the tags that you want to populate with data in the first row of the table with data from specific columns.
MappingDateTime	Select the time reference: UTC or Local.
Description	Enter a description for the table configuration.
[Other columns]	For definitions of other columns that are available in this table, see Common Column Definitions.

- Continue adding as many table configurations as you need.

Reading and Writing the Table Contents

Runtime access for the table contents is executed automatically when the table is mapped to a DataGrid object. See Configuring a DataGrid Window. You can also get the table contents or perform operations on the tables, using the runtime properties for the Dataset table object. See http://www.tatsoft.com/help/fs-2014/runtime/index.html.

Configuring Database Queries

You can configure queries to perform more advanced functions with SQL statements to work with data from external databases.

To configure database queries:

- Go to **Edit >Datasets** > **Queries**.
- Enter or select information, as needed.

Column	Description
Name	Enter a name for the query. The system lets you know if the name is not valid.
DB	Select the database configuration.
SqlStatement	Enter the query using SQL syntax.
Mapping	Click ... to select the tags that you want to populate with data from specific columns returned by the query.
MappingDateTime	Select the time reference: UTC or Local.
Description	Enter a description for the table configuration.
[Other columns]	For definitions of other columns that are available in this table, see Common Column Definitions.

- Continue adding as many queries as you need.

Getting the Query Contents

Runtime access for the table contents is executed automatically when the query is mapped to a DataGrid object. See Configuring a DataGrid Window.

You can also get the query contents or perform operations on the query, using the runtime properties for the Dataset.Query object. See http://www.tatsoft.com/help/fs-2018/runtime/index.html.

Configuring Files for Data Exchange

You can configure files to retrieve data from a network location.

To configure database files:

- Go to **Edit > Datasets** > **Files**.
- Enter or select information, as needed.

Column	Description
Name	Enter a name for the file configuration. The system lets you know if the name is not valid.
FileName	Enter the full path to the file.
FileType	Select the type of file.
Objects	Click ... to select the tags that you want to populate with data from the file with data from specific columns.
Description	Enter a description for the file configuration.
XMLSchemaType	Represents the schema type of an XML file, which can be: a TagList: XML that contains a tag list with the tag name and tag value or a TagObject: XML that contains the entire tag tree and its children.
[Other columns]	For definitions of other columns that are available in this table, see Common Column Definitions.

- Continue adding as many file configurations as you need.

The Dataset Namespace

The namespace **Dataset** is the entry point to all objects related to the Datasets module.

The **Dataset.DB** object lists all configured database connections and its runtime properties

The **Dataset.Table** object lists all configured tables and their runtime properties

The **Dataset.Query** object lists the defined queries and their runtime properties.

The **Dataset.File** object lists the defined queries and their runtime properties.

See Namespaces for the complete programming reference on runtime objects.