

Creating Data Markers

A **data marker** is a cell value or sheet tab name beginning with `%%=` or `%%=$` that specifies a database column, a variable, or an array to insert in the spreadsheet column or sheet tab containing the marker. Data markers are created in Microsoft Excel and bound to data sources in code. When you run the code, ExcelWriter populates the data markers with values from a data source.

Refer to the [ExcelTemplate Code Samples](#) to see live demonstrations of various ExcelTemplate applications. Click the "View template" link on each sample page to see the template and view the Data Markers.



Using data markers in tab sheet names was introduced in ExcelWriter 8.5.0. For more information, please refer to [Using Data Markers in Sheet Tabs](#).

A data marker binds in script to a **data source** which may be a variable, an array, or an ADO.NET DataSet, DataTable, or DataReader. A data marker may include [modifiers](#). Data source and field numbers are 1-based. If ExcelWriter encounters `%%=#0[field]` or `%%=[DataSource.]#0`, an error will occur.



In ExcelWriter versions before 3.1, data source and field numbers were 0-based. If you upgraded from an earlier version, you may need to modify your code.

Implicitly or explicitly, a data marker always specifies both a **data source** and a **column** - even if the data source is a variable or a one-dimensional array. For example, in the data marker `%%=Products.ProductID`, "Products" is the data source, and "ProductID" is the column. The data marker `%%=ProductID` means the "ProductID" column from the first data source defined in the ExcelWriter code.

Both the data source and the column can be specified by either name or number:

<code>%%=Products.ProductID</code>	The "ProductID" column of the data source "Products."
<code>%%=#2.FirstName</code>	The "FirstName" column of the second data source defined in the ExcelWriter code.
<code>%%=Employee.#3</code>	The third column of the "Employee" data source.
<code>%%=#2.#3</code>	The third column of the second data source defined in the ExcelWriter code.

- [Data Marker Names](#)
- [Data Marker Formats](#)
- [Short Data Markers](#)
- [Long Data Markers](#)
- [Grouping and Nesting Markers](#)
- [Data Marker Modifiers](#)

Data Marker Names

Follow these rules when naming data markers:

- Data source and column names must not include Unicode characters unless using brackets.
- Data source and column names must begin with a letter (A-Z, a-z).
- Data source and column names may include the following characters only without using brackets: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890_
- Spaces are not allowed anywhere in a data marker except within a column name or data source in brackets (for example, `%%=Products.[Product Category]`).

To include spaces or Unicode characters in the data source column name, use this format: `%%=DataSource.[Column Name]`

For example: `%%=Employees.[Street Address]`

The [Column Name] format allows you to match any SQL column name exactly. Legal characters within the brackets include any Unicode character. Additionally, data source names in the [Data Source] format can include any Unicode character.

Data Marker Formats

Data Marker Formats		
For this data source:	And this method:	Use one of these data marker formats:

DataSet DataTable DataReader	BindData SetDataSource(deprecated)	%%=DataSourceName.ColumnName[(modifier)] %%=#DataSourceNumber.#ColumnNumber[(modifier)] %%=DataSourceName.#ColumnNumber[(modifier)] %%=#DataSourceNumber.ColumnName[(modifier)] %%=ColumnName[(modifier)] %%=#ColumnNumber[(modifier)]
Variable	BindCellData SetCellDataSource(deprecated)	%%=\$DataSourceName[(modifier)] %%=\$#DataSourceNumber[(modifier)]
1-Dimensional Array	BindColumnData SetColumnDataSource(deprecated)	%%=\$DataSourceName[(modifier)] %%=\$#DataSourceNumber[(modifier)]
1-Dimensional Array	BindRowData SetRowDataSource(deprecated)	%%=DataSourceName.ColumnName[(modifier)] %%=#DataSourceNumber.#ColumnNumber[(modifier)] %%=DataSourceName.#ColumnNumber[(modifier)] %%=#DataSourceNumber.ColumnName[(modifier)] %%=ColumnName[(modifier)] %%=#ColumnNumber[(modifier)]
2-dimensional Array	BindData SetDataSource(deprecated)	%%=DataSourceName.ColumnName[(modifier)] %%=#DataSourceNumber.#ColumnNumber[(modifier)] %%=DataSourceName.#ColumnNumber[(modifier)] %%=#DataSourceNumber.ColumnName[(modifier)] %%=ColumnName[(modifier)] %%=#ColumnNumber[(modifier)]



In ExcelWriter 4, a \$ was required for all data markers that bind to 1-dimensional arrays. In ExcelWriter 5 and above, if a 1-dimensional array binds to a row of data markers by the **ExcelTemplate.SetRowDataSource** or [ExcelTemplate.BindRowData](#) methods, the \$ should be omitted. However, for backward compatibility, if you include the \$, the data marker will be populated without error.

Using Short Data Markers

ExcelWriter supports several full-length and short data marker forms. When you use any of the long data marker formats, you explicitly include both the data source name or number and the column name or number. Short formats do not specify both data source and column explicitly: one or the other is omitted. How ExcelWriter understands a short data marker depends on whether or not the marker contains a \$.

ExcelWriter understands %%=\$**Employee** as the first column of the data source "Employee." If you omit the \$, ExcelWriter will read "Employee" as a column from the first data source defined in the ExcelWriter code. So:

- %%=\$**Employee** means the first column in the data source "Employee."
- %%=**Employee** means the "Employee" column in the first data source defined in the ExcelWriter code.

Examples

Form	Example
%%=#ColumnNumber	%%=#7 Seventh column of the first data source defined in the ExcelWriter code.
%%=\$#DataSourceNumber	%%=\$#7 First column of the seventh data source defined in the ExcelWriter code.
%%=ColumnName	%%= OrderId "OrderId" column from the first data source defined in the ExcelWriter code.

%%=\$DataSourceName	%%=\$Orders First column in the data source "Orders."
---------------------	---

Long Data Markers

When you use any of the long data marker formats, explicitly include both the data source name or number and the column name or number. Using the long formats may prevent confusion when writing the ExcelWriter code, since you must be able to distinguish between data source and column to set the data source in code.

Examples

Form	Example
%%=DataSourceName.ColumnName	%%=Employees.FirstName "FirstName" column in the "Employees" data source.
%%=DataSourceName.#ColumnName	%%=Products.#3 Third column of the data source "Products."
%%=#DataSourceNumber.ColumnName	%%=#2.ProductId "ProductId" column of the second data source defined in the ExcelWriter code.
%%=#DataSourceNumber.#ColumnName	%%=#3.#2 Second column of the third data source defined in the ExcelWriter code.
%%=DataSourceName.[Column Name]	%%=Employees.[Street Address] "Street Address" column of the data source "Employees."

Grouping and Nesting Markers

(Enterprise Edition only)

In addition to data markers, ExcelTemplate uses Grouping and Nesting markers to format flat data as it is brought into a spreadsheet. The **%%group**, **%%endgroup**, **%%header**, and **%%footer** markers are used to signify the beginning and ending of the group formatting and the header and footer sections; their rows will not be included in the worksheet. The **%%value(data marker)** marker allows you to include a data marker value in the header or footer of a group, while the **(hide)** modifier, described below, allows you to remove a field from the data marker rows if its values are repeated in the header and footer rows.



Grouping and Nesting Markers are only evaluated in Office Open XML (.xlsx) files. Grouping and Nesting Markers placed in BIFF8 (.xls) templates will not be evaluated.

Grouping and Nesting Marker Descriptions

(Enterprise Edition only)

%%group	Signifies the beginning of a grouping block. Place this marker in the column of the field by which you wish to group. Grouping and Nesting formatting will be applied to the rows from this marker to the %%endgroup marker. This row will not be displayed in the output spreadsheet. The group marker is required for grouping functionality to work.
%%header	Signifies the beginning of the header rows for a particular column in a grouping block. This marker should be placed in a row directly under the row containing the %%group markers or other header rows and above the data marker row in the column. All rows between this marker's row and the data marker row, or the next header marker, will be repeated for each new value in the grouping field. The row containing the %%header marker will not be displayed in the output spreadsheet. Header markers are optional.
%%value(data marker)	The %%value(data marker) marker is used to include the values of a data marker row in the header or footer of a group. The current value of the field whose data marker is placed as a modifier for this marker will be brought into the header or footer, replacing the %%value() marker. Value markers are optional.

%%footer	Placed below the data marker rows, this marker signifies the beginning of the footer rows for a particular column in a grouping block. All rows between this marker's row and the %%endgroup marker row or the next footer marker will be repeated once for each new value in the grouping field. The row containing the %%footer marker will not be displayed in the output spreadsheet. Footer markers are optional.
%%endgroup	Signifies the end of a grouping block. Place this marker in the same column as the %%group marker of the group you are ending. The row containing the %%footer marker will not be displayed in the output spreadsheet. Only one endgroup marker is required, no matter how many group markers are in the first row of the grouping block.

Example

Below is an example of a grouping block in a template spreadsheet.

%%group				
%%header				
%%value(data.#1)				
%%=data.#1(hide)	%%=data.#2	%%=data.#3	%%=data.#4	%%=data.#5
%%footer				
Subtotal row				
%%endgroup				

For the data to be grouped correctly, they must be in a flat format and sorted hierarchically by grouping categories. For the grouping block above, for example, you could use the following data table:

First category	first row	2	4	3
First category	second row	3	4	7
First category	third row	3	5	2
Second category	first row	5	2	9
Second category	second row	4	3	1
Second category	third row	8	5	5

The [ExcelTemplate.BindData](#) method should bring the data in as displayed below:

First category				
	first row	2	4	3
	second row	3	4	7
	third row	3	5	2
Subtotal row				
Second category				
	first row	5	2	9
	second row	4	3	1
	third row	8	5	5
Subtotal row				


Data Marker Modifiers

An ExcelWriter data marker can include modifiers, included in parentheses at the end of the data marker, that change the behavior of ExcelTemplate. For example:

%%=Products.ProductID(Lowercase)

If a data marker includes more than one modifier, the modifiers should be separated by commas. Do not include spaces between modifiers.

ExcelWriter supports the following data marker modifiers:

Fieldname	Use the Fieldname modifier to insert database column headings in a spreadsheet. For example, if template cell C5 contains %%=Products.#2(fieldname) ExcelWriter will insert the name of column 2 in C5 .
Uppercase	If a data marker includes the Uppercase modifier, all text values in the data marker column or cell will be displayed in uppercase. If a data marker includes both the Uppercase and Lowercase modifiers, the last will be applied to cell values.
Lowercase	If a data marker includes the Lowercase modifier, all text values in the data marker column or cell will be displayed in lowercase. If a data marker includes both the Uppercase and Lowercase modifiers, the last will be applied to cell values.
Optional	<p>By default, if a data marker's data source name or column name is not bound to a data source in the script, an error will occur. If a data marker contains the Optional modifier, and the data marker is not bound by name to a data source, ExcelWriter will discard the data marker in the generated spreadsheet and will not throw an error. This modifier makes both a data marker's data source and column optional. For example, for this data marker:</p> <p>%%=Orders.OrderID(Optional)</p> <p>If either the data source "Orders," or the column "OrderId" does not exist, the data marker will be removed without error.</p> <div><ul style="list-style-type: none">• If a data marker contains a data source number and/or field number (for example %%=#7.#3) and is not bound to a data source in script, the data marker will be removed without error whether the "Optional" modifier is present or not.• If the RemoveExtraDataMarkers property is set to true, all data markers that do not bind to data sources will be removed without error.</div>
Preserve	The Preserve modifier forces Excel to always insert a string value as a string, instead of attempting to convert the value to a number.
Convert	The Convert modifier forces Excel to always try converting a string to a number. If the conversion fails, the value will be inserted as a string.
Continue	The Continue modifier is used when the user wishes to bind data from a single data source into multiple worksheets. If the data source for the ExcelTemplate uses a forward-only cursor, such as with a DataReader, the second worksheet with the same data marker will automatically start from the row after the last row in the first worksheet. If the data source is scrollable, however, the second worksheet with the same data marker will start from the first row of the data source. The Continue modifier tells the ExcelTemplate object to start from the row after the last row in the previous worksheet when a scrollable data source is being used.
Hide	The Hide modifier removes the field from the spreadsheet after data is bound to the spreadsheet. This is usually used to allow you to remove field values if they are already expressed in a Group header or footer using the %%value() marker. Since the data are removed on the server side, they will not be evaluated in any formulas on the spreadsheet.



The (hide) data marker modifier will only be evaluated in Office Open XML (.xlsx) files. (hide) data marker modifiers in BIFF8 (.xls) files will not be evaluated and may throw an exception.