

Creating Merge Fields

Table of Contents

- [Merge Fields](#)
- [Merge Field Names](#)
- [Using Short Merge Fields](#)
 - [Examples](#)
- [Using Long Merge Fields](#)
 - [Examples](#)
- [Merge Field Modifiers](#)
- [Merge Field Formulas](#)

Merge Fields

A WordWriter template is a Microsoft Word file that contains merge fields. A merge field displays a data source field (for example, a database column name). A merge field is created in Microsoft Word and bound in code to a data source. The data source may be an array, a DataSet, a DataTable, or a DataReader. When you run the code, WordWriter populates the merge fields with data source values.

A template may contain multiple sets of merge fields. Each set of fields binds to a single data source. Implicitly or explicitly, a merge field always specifies both a **data source** and a **column**. For example, in the merge field*«Products.ProductID», "**Products**" is the data source and "**ProductID**" is the column. The merge field«ProductID»* refers to the "ProductID" column from the first data source defined in the WordWriter code.

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

«Products.ProductID»	The "ProductID" column of the data source "Products."
«#2.FirstName»	The "FirstName" column of the second data source defined in the WordWriter code.
«Employee.#3»	The third column of the "Employee" data source.
«#2.#3»	The third column of the second data source defined in the WordWriter code.

Data source and field numbers are 1-based. If WordWriter encounters «#0[field]» or «[DataSource.]#0», an error will occur.

The default data source for a main document merge field is the first data source set in the WordWriter code (by the method SetDataSource). The default data source for a merge field within a repeat block is the data source set for the repeat block (by the method SetRepeatBlock, which specifies the repeat block by bookmark name). You can include main document merge fields within a repeat block, however, these fields will not be repeated when the template is populated (only one row of data can be imported to main document merge fields). The data source for a repeat block is only valid within the repeat block; you cannot include a repeat block merge field outside of its bookmark-defined repeat block.



The Main Document and Repeat Blocks

A repeat block is a text fragment, a list, or a table row - defined by a Word bookmark - that contains merge fields and that will be repeated for each row returned from the database. The main document is any part of the template that is not within a repeat block. You can assign one data source to each repeat block, and multiple data sources to the main document. Only one row of data can be imported to each set on main document merge fields.

- [Merge Fields Names](#)
- [Using Short Merge Fields](#)
- [Using Long Merge Fields](#)
- [Merge Field Modifiers](#)
- [Merge Field Formulas](#)

Merge Field Names

Follow these rules when naming merge fields:

- Data source and column names must not include Unicode characters.
- 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:
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890_



If you assigned a different data source separator, you can also use a "." in your data source and column names.

- Spaces are not allowed anywhere in a merge field.



Use brackets if your column name does not conform to these rules

The exception of these rules is when a column name is enclosed 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 every character except parentheses () or brackets [].

When creating these fields with WordApplication or editing the field codes manually, the merge field name must be enclosed in quotes. For example, it should be { MERGEFIELD "[data source].[column name]" }. Any modifiers must also go in the quotes, but the formatting options must go after the quotes.

Using Short Merge Fields

WordWriter supports several full-length and short merge field forms. When you use any of the long merge field 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 WordWriter understands a short merge field depends on whether or not the marker contains a \$.

WordWriter understands «\$Employee» as the first column of the data source "Employee." If you omit the \$, WordWriter will read "Employee" as a column from the first data source defined in the WordWriter code.

- «\$Employee» means the first column in the data source "Employee."
- «Employee» means the "Employee" column in the first data source defined in the WordWriter code.

Examples

Form	Example
«#ColumnNumber»	«#7» Seventh column of the first data source defined in the WordWriter code.
«\$#DataSourceNumber»	«\$#7» First column of the seventh data source defined in the WordWriter code.
«ColumnName»	«OrderId» "OrderId" column from the first data source defined in the WordWriter code.
«\$DataSourceName»	«\$Orders» First column in the data source "Orders."



If you are using the [WordTemplate.SetMailMerge](#) method, you **must** use Short Merge fields.

Using Long Merge Fields

When you use any of the long merge field formats, you 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 WordWriter code, since you must be able to distinguish between data source and column to set a data source in code.

Examples

Form	Example
«DataSourceName.ColumnName»	«Employees.FirstName» "FirstName" column in the "Employees" data source.
«DataSourceName.#ColumnNumber»	«Products.#3» Third column of the data source "Products."
«#DataSourceNumber.ColumnName»	«#2.ProductId» "ProductId" column of the second data source defined in the WordWriter code.

«#DataSourceNumber.#ColumnName»	«#3.#2» Second column of the third data source defined in the WordWriter code.
«DataSourceName.[Column Name]»	«Employees.[Street Address]» "Street Address" column of the data source "Employees."

Merge Field Modifiers


An WordWriter merge field can include modifiers. Modifiers are included in parenthesis at the end of the merge field. For example:

```
<<Products.ProductID(Lowercase)>>
```

If a merge field includes more than one modifier, the modifiers should be separated by commas. Do not include spaces between modifiers. For example:

```
<<Products.ProductID(Fieldname,Lowercase,Optional)>>
```

WordWriter supports the following merge field modifiers:

Fieldname	Use the Fieldname modifier to insert database column headings in a document. For example, in place of the merge field «RecordSet.#2(fieldname)», WordWriter will insert the name of the second column in the RecordSet.
Uppercase	If a merge field includes the Uppercase modifier, all text values imported to the merge field will be displayed in uppercase. If a merge field includes both the Uppercase and Lowercase modifiers, the last will be applied.
Lowercase	If a merge field includes the Lowercase modifier, all text values imported to the merge field will be displayed in lowercase. If a merge field includes both the Uppercase and Lowercase modifiers, the last will be applied.
Optional	By default, if a merge field's data source name or column name is not bound to a data source in code, an error will occur. If a merge field contains the Optional modifier, and the merge field is not bound by name to a data source, WordWriter will discard the merge field in the generated document and will not throw an error. This modifier makes both a merge field's data source and column optional. For example, for this merge field: «Orders.OrderID(Optional)» If either the data source "Orders," or the column "OrderId" does not exist, the merge field will be removed without error. <div style="border: 1px solid yellow; padding: 5px; margin-top: 10px;"> If a merge field contains a data source number and/or field number (for example «#7.#3») and is not bound to a data source in script, the merge field will be removed without error whether the "Optional" modifier is present or not.</div>
Image	If the image modifier is present and the column of data for the merge field is either of type byte array (byte[]) or SQL BLOB, then the image will be rendered. The image modifier can take optional parameters to format the size of the image (e.g. <<Product.Image(image(1,600,600))>>. For more see Inserting an Image .
Document(format)	The document modifier can be used to insert certain types of documents into docx files where format is either rtf, html, or docx. This will insert the formatted document into the docx output. For more see Inserting an Embedded Document .

Merge Field Formulas



Word formulas are only supported with the Word 2007 OOXML file format (.docx/.docm).

Beginning with WordWriter 4.1, the WordTemplate object evaluates a number of different formulas placed in merge fields using the format «=FORMULA(DataSource.Field)». Each of the formulas will perform a specific operation on the values of the referenced data field. Formulas are evaluated for all values of the field in the data source, regardless of how many of the values are actually displayed in the document.

WordWriter supports the following merge field formulas:

AVERAGE	Calculates the average of the field values
COUNT	Returns the number of values (that is, the number of rows in the data source) of the field
COUNTA	Returns the number of non-null values of the field
MAX	Returns the maximum value from the field
MIN	Returns the minimum value from the field
PRODUCT	Calculates the product of the field values
STDEV	Returns the standard deviation of the field values, treating the values as a sample
STDEVP	Returns the standard deviation of the field values, treating the values as a population
SUM	Calculates the sum of the field values
VAR	Returns the variance of the field values, treating the values as a sample
VARP	Returns the variance of the field values, treating the values as a population