

Hello World using WordTemplate

Table of Contents

- Hello World with WordTemplate
- Setting up the template file
- Writing the Code
- Final Code
- Downloads

Hello World with WordTemplate

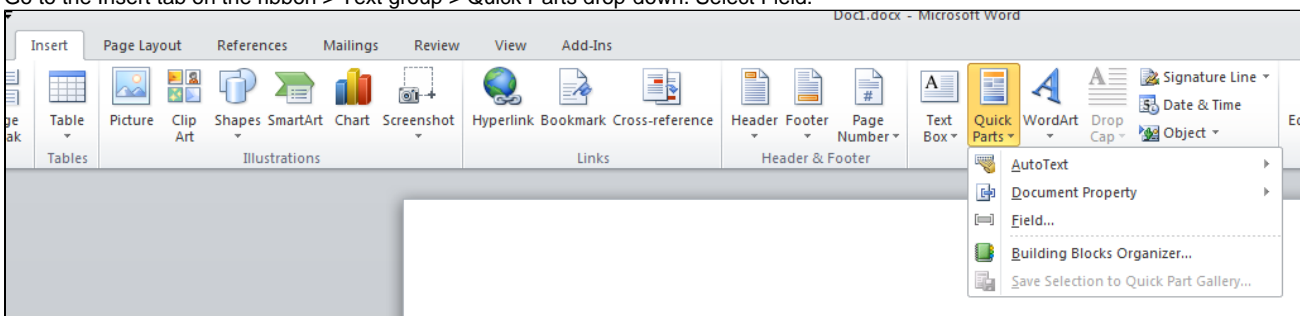
WordWriter's WordTemplate approach allows you to write data to a template file that contains merge field. The merge fields tell WordWriter where to bind specific sets of data, similar to Word's mail merge. This tutorial will show you the basics on how to dynamically insert data into a document using WordTemplate by taking custom text from a web form textbox and inserting it into a template file.

Setting up the template file

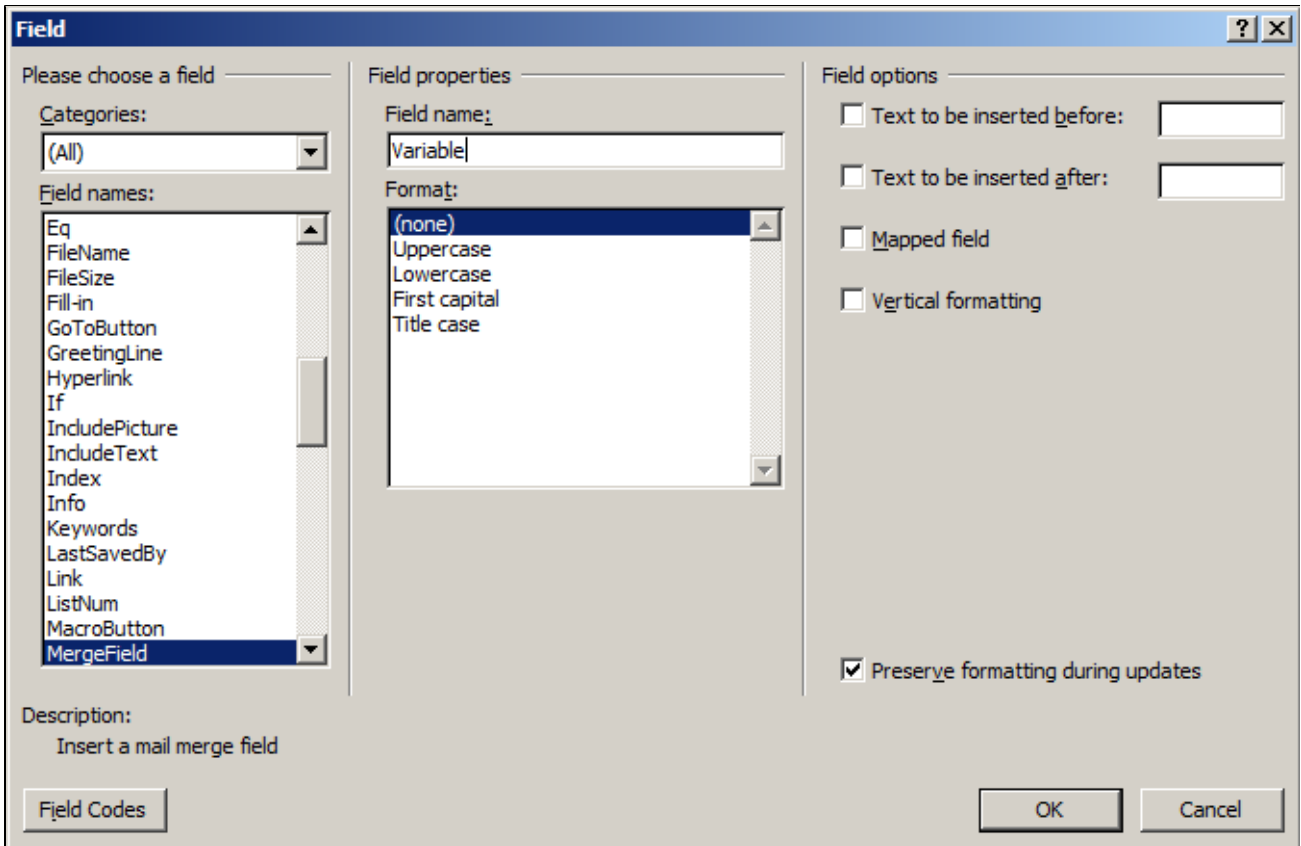
1. Create a new .docx file. Save it as *template.docx*.

 In the sample code, the completed template file is located in templates/template.docx

2. Go to the Insert tab on the ribbon > Text group > Quick Parts drop-down. Select Field.

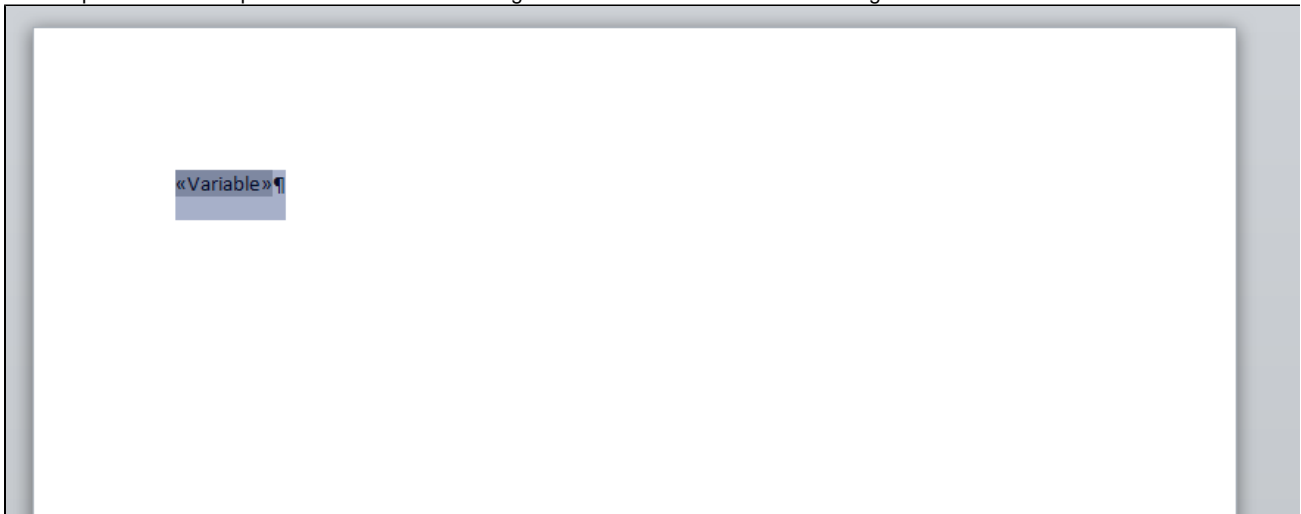


3. In the Field dialog, select Mergefield from the Field names list.
4. In the Field Properties section, type "Variable" in the Field name box. Click OK.




This will create a merge field <<Variable>> which will correspond to a column in a data set with only one row of data.

5. The template is now complete. We'll move on to writing the code to bind the data to this merge field.



Writing the Code

 In the sample code, the sample web application page `WorldTemplate_HelloWorld.aspx` and code behind `WorldTemplate_HelloWorld.aspx.cs/vb` are available in the Hello World project.

1. Include the *SoftArtisans.OfficeWriter.WordWriter* namespace in the code behind.

```
using SoftArtisans.OfficeWriter.WordWriter;
```

```
Imports SoftArtisans.OfficeWriter.WordWriter
```

2. Instantiate the *WordTemplate* object.

```
WordTemplate WT = new WordTemplate();
```

```
Dim WT As WordTemplate = New WordTemplate()
```

3. Open the template file with *WordTemplate.Open*

```
WT.Open(Page.MapPath("templates\\template.docx"));
```

```
WT.Open(Page.MapPath("templates\\template.docx"))
```

4. In this example, we'll pull a text value from a textbox on a web form that a user submitted. Get the data value from the web form's text box

```
string value = DataValueBox.Text.Trim();
```

```
Dim value As String = DataValueBox.Text.Trim()
```

5. Create an object array to hold the textbox value. Create a string array to hold the column name "Variable"

```
object[] values = { value };  
string[] columnNames = { "Variable" }
```

```
Dim values() As Object = {value}  
Dim columnNames() As String = {"Variable"}
```

6. Use `WordTemplate.SetDataSource` to bind the data to template file.

```
WT.SetDataSource(values, columnNames);
```

```
WT.SetDataSource(values, columnNames)
```

`WordTemplate` has three methods to bind data: `WordTemplate.SetDataSource`, `WordTemplate.SetRepeatBlock`, and `WordTemplate.SetMailMerge`. `SetDataSource` binds a single row of data to the template, where the merge fields can span the entire document. To bind a single value or group of single values, you need to put those in a data set, such as an array and then bind that array to the template file.

7. Call `WordTemplate.Process()` to bind the data to the template file.

```
WT.Process();
```

```
WT.Process()
```

8. Save the output with `WordTemplate.Save`

```
WT.Save(Response, "Output.docx", false);
```

```
WT.Save(Response, "Output.docx", False)
```

There are several options for `WordTemplate.Save` including: save to disk, save to memory stream, stream back to the client inline, and stream back to the client as an attachment. In this case, we're streaming the document back to the client as an attachment.



WordWriter does not convert between file formats, so it is important that the file extension on the output file matches the file extension of the original template file.

9. Run your code.

In the output file you will see that the merge field has been replaced by the value.



Congratulations, you completed *Hello World using WordTemplate!*

Final Code

```
using
SoftArtisans.OfficeWriter.WordWriter;
...
WordTemplate WT = new WordTemplate();

WT.Open(Page.MapPath("templates\\templa
te.docx"));

string value =
DataValueBox.Text.Trim();

object[] values = { value };
string[] columnNames = { "Variable" };

WT.SetDataSource(values, columnNames);

WT.Process();

WT.Save(Response, "Output.docx",
false);
```

```
Imports
SoftArtisans.OfficeWriter.WordWriter
...
Dim WT As WordTemplate = New
WordTemplate()

WT.Open(Page.MapPath("templates\\templa
te.docx"))

Dim value As String =
DataValueBox.Text.Trim()

Dim values() As Object = {value}
Dim columnNames() As String =
{"Variable"}

WT.SetDataSource(values, columnNames)

WT.Process()

WT.Save(Response, "Output.docx", False)
```

Downloads

You can download the code for the Hello World tutorial as a Visual Studio solution.

- [\[WordWriter_HelloWorldC#.zip\]](#)
- [\[WordWriter_HelloWorldVB.zip\]](#)