

# Hello World using WordApplication

## Table of Contents

- [Diving right into the code](#)
- [Downloads](#)

## Hello World with ExcelApplication

WordWriter's WordApplication approach provides you with full programmatic control over the binary Word file formats (DOC, DOT). This includes the ability to create and modify: paragraphs, tables, lists, hyperlinks, bookmarks, images and more! This tutorial will show you how to create a new document and write some text to that document.

## Diving right into the code



In the Hello World sample web application, WordApplication\_HelloWorld.aspx has a text box for users to supply a value and WordApplication\_HelloWorld.aspx.cs/vb contains the code show below.

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

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

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

2. Instantiate the [WordApplication](#) object

```
WordApplication WAPP = new WordApplication();
```

```
Dim WAPP As WordApplication = new WordApplication()
```

3. Create a new document with [WordApplication.Create\(\)](#)

```
Document DOC = WAPP.Create();
```

```
Dim DOC As Document = WAPP.Create()
```

Unlike the WordTemplate object, which represents a single file, WordApplication works as a file generation engine. The WordApplication object can be used to create, open, and save multiple documents at once.



WordApplication only supports the Office 2003 file formats - DOC, DOT. WordWriter does not currently support creating DOCX/DOCM files with WordApplication.

4. For this example, we'll pull a value from a text box in a web form. Get the value.

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

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

5. Insert the text into the document with [Element.InsertTextAfter](#).

```
DOC.InsertTextAfter(value, true);
```

```
DOC.InsertTextAfter(value, true)
```

Text strings need to be inserted into a paragraph, table cell, or list entry elements in a document. In this case, when the text is inserted into the document, it's actually inserted into the last paragraph in the document. Documents created by WordApplication come with a paragraph, so we do not need to create that paragraph.

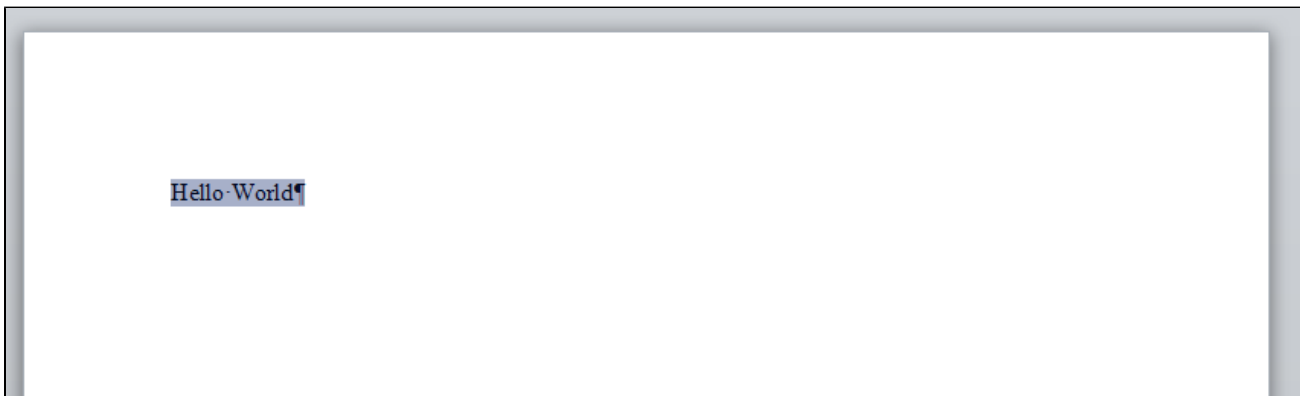
6. Save the output file with [WordApplication.Save](#)

```
WAPP.Save(DOC, Response, "Output.doc", false);
```

```
WAPP.Save(DOC, Response, "Output.doc", false)
```

WordApplication.Save has the same output options as WordTemplate: 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.

Here is an example of what the output should look like:



Congratulations, you have completed *Hello World using WordApplication*!

```
using
SoftArtisans.OfficeWriter.WordWriter;
...
WordApplication WAPP = new
WordApplication();

Document DOC = WAPP.Create();

string value =
DataValueBox.Text.Trim();

DOC.InsertTextAfter(value, true);

WAPP.Save(DOC, Response, "Output.doc",
false);
```

```
Imports
SoftArtisans.OfficeWriter.WordWriter
...
Dim WAPP As WordApplication = new
WordApplication()

Dim DOC As Document= WAPP.Create()

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

DOC.InsertTextAfter(value, true)

WAPP.Save(DOC, Response, "Output.doc",
false)
```

## Downloads

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

- [\[WordWriter\\_HelloWorldC#.zip\]](#)
- [\[WordWriter\\_HelloWorldVB.zip\]](#)