

WordTemplate Output Options

Table of Contents

- [Save to Disk](#)
- [Open in the Browser](#)
- [Open in the Default Application for .doc or .docx Files](#)
- [Save and Stream to Browser](#)
- [Pass the Template to WordApplication](#)
 - [Example](#)

After populating your WordWriter template from a data source, call Save to either save the new Word file on the server, or stream it to the browser. WordWriter can:

- [Save the generated Word file to disk .](#)
- [Stream the Word file to the client and open it in the browser window .](#)
- [Stream the Word file to the client and open it in the default application for .doc and .docx Files .](#)
- [Save the generated Word file on the server and stream it to the client .](#)
- [Pass the template to WordApplication .](#)

Save to Disk

To save the generated file on the server:

Pass the Save method a file path, for example:

C#

```
WordTemplate wt = new WordTemplate();
wt.Process();
wt.Save(@"c:\Documents\File.doc");
```

VB.NET

```
Dim wt As New WordTemplate()
wt.Process()
wt.Save("c:\Documents\File.doc")
```

OR

Pass the Save method a System.IO.FileStream object, for example:

C#

```
using System.IO; // For FileStream object
WordTemplate wt = new WordTemplate();
wt.Process();
using (FileStream fstream = new FileStream(@"c:\Documents\File.doc",
    FileMode.Create))
{
    wt.Save(fstream);
    fstream.Close();
}
```

VB.NET

```
Imports System.IO ' For FileStream object
Dim wt As New WordTemplate()
wt.Process()
Dim fstream As new FileStream("c:\Documents\File.doc",
    FileMode.Create)
Try
    wt.Save(fstream)
    fstream.Close()
Finally
    If Not fstream Is Nothing Then
        fstream.Dispose()
    End If
End Try
```

Open in the Browser

When you pass an `HttpResponse` object to `Save`, `WordWriter` will stream the generated Word file to the client. The browser will display a File Download dialog asking the user to open or save the file. The method's second parameter specifies a file name to display in the File Download dialog. If the method's third parameter - `OpenInBrowser` - is `True` and the user chooses to open the Word file, the file will open in the browser window.

C#

```
wt.Save(Page.Response, "StringVarOutput.doc", true);
```

VB.NET

```
wt.Save(Page.Response, "StringVarOutput.doc", True)
```

Alternatively, stream the file to the browser by passing `Response.Stream` to the `Save` method. In this case, to open the generated file in the browser window:

1. Set the response's content-type header to "application/vnd.ms-word":

```
Response.ContentType = "application/vnd.ms-word"
```

2. Set the response's content-disposition to "inline":

```
Response.AddHeader("Content-Disposition", "inline;filename=""WordWriter.doc"")
```

3. Pass the `Save` method `Response.Stream` as a parameter:

```
oWW.Save(Response.OutputStream)
```

Open in the Default Application for .doc or .docx Files

When you pass an `HttpResponse` object to `Save`, `WordWriter` will stream the generated Word file to the client. The browser will display a File Download dialog asking the user to open or save the file. The method's second parameter specifies a file name to display in the File Download dialog. If the method's third parameter - `OpenInBrowser` - is `False` and the user chooses to open the Word file, the file will open in the default application for .doc or .docx files (such as Microsoft Word, StarOffice, or OpenOffice.org).

C#

```
bool bOpenInBrowser = false;
wt.Save(Page.Response, "StringVarOutput.doc", bOpenInBrowser);
```

VB.NET

```
Dim bOpenInBrowser As Boolean = False
wt.Save(Page.Response, "StringVarOutput.doc", bOpenInBrowser)
```

Alternatively, stream the file to the browser by passing `Response.Stream` to the `Save` method. In this case, to open the generated file in the default application for Word files:

1. Set the response's content-type header to "application/vnd.ms-word":

C#

```
Response.ContentType = "application/vnd.ms-word";
```

VB.NET

```
Response.ContentType = "application/vnd.ms-word"}}
```

2. Set the response's content-disposition to "attachment":

C#

```
Response.AddHeader("Content-Disposition",
    "attachment;filename=\"/WordWriter.doc/\"");
```

VB.NET

```
Response.AddHeader("Content-Disposition", \_
    "attachment;filename=\"WordWriter.doc\"")
```

3. Pass the `Save` method `Response.Stream` as a parameter:

C#

```
wt.Save(Response.OutputStream);
```

VB.NET

```
wt.Save(Response.OutputStream)
```

Save and Stream to Browser

You can call Save more than once for a single instance of WordTemplate. In the following examples Save is called twice:

```
1. oWW.Save(Server.MapPath("./SaveMultipleOutput.doc"))
```

Saves the generated Word file on the server.

```
2. oWW.Save(Page.Response, "SaveMultipleOutput.doc", False)
```

Streams the generated Word file to the client. Since the parameter OpenInBrowser is False, the file will open in the default application for .doc files.

Pass the Template to WordApplication

You can use WordTemplate to open and populate a WordWriter template, then pass the populated workbook to WordApplication for additional processing. In this case, do not call Save. Instead, pass the WordTemplate object to WordApplication's Open method:

```
Document doc WordApplication.Open(WordTemplate template);
```

Example

C#

```
WordTemplate wTempl = new WordTemplate();  
wTempl.Process();  
  
WordApplication wApp = new WordApplication();  
Document doc = wApp.Open(wTempl);
```

VB.NET

```
Dim wTempl As New WordTemplate  
wTempl.Process()
```

```
Dim wApp As New WordApplication  
Dim doc As Document = wApp.Open(wTempl)
```