Version 2.0.2.0  
egonl  
May 2020

SharpDocx

Summary

Generating documents with SharpDocx is a two-step process. First you create a view in Word. A view is a Word document which also contains C# code. Code can be inserted anywhere, e.g. 28-5-2020 00:09:20 would insert the current date and time.

The next step is to create documents based on this view. This requires two lines of code:

var document = DocumentFactory.Create("view.cs.docx");

document.Generate("output.docx");

Out of the box SharpDocx supports inserting text, tables, images and more. This tutorial shows you how.

If you want, you can specify a view model to be used in your view. Then you could write things like < % foreach (var item in Model.MyList) { % >. See the Model sample.

If you want to do something that's not supported by SharpDocx, you can do so by creating your own document subclass. See the Inheritance example.

SharpDocx is inspired by Web technologies like ASP.NET and JSP. Developers familiar with those technologies should feel right at home.

Generating documents with SharpDocx can be very fast: a slightly modified Model sample produced 25 documents per second on my modest laptop. That’s 1500 documents per minute. Single threaded.

Contents

[The basics 3](#_Toc32010234)

[The Write method 3](#_Toc32010235)

[Conditional content 4](#_Toc32010236)

[Loops 5](#_Toc32010237)

[Nested loops 6](#_Toc32010238)

[Loops, tables and the AppendRow method 7](#_Toc32010239)

[Combining loops, text blocks and tables 8](#_Toc32010240)

[Images 9](#_Toc32010241)

[Replacing text 10](#_Toc32010242)

[Referencing assemblies and importing namespaces 11](#_Toc32010243)

[The Map 12](#_Toc32010244)

[The SharpDocx solution 13](#_Toc32010245)

# The basics

At any point in the text you can insert C# statements. Like right here.

The result looks like this:



## The Write method

If you want to display the value of i, you can use the Write method. Right now, i is 1.

This will show:



There’s also a shorthand notation for the Write method: i is still 1.

This results in:



You can insert line breaks by using ‘\n’:

This paragraph  
contains two  
line breaks.

# Conditional content

You can use an if statement to display conditional content.

This will be displayed.

In this case, any formatting will be lost because the code parser ignores any formatting.

If you want to conditionally display a paragraph with formatting, use a *text block*: text between two code blocks and placed between curly brackets:

This **will** *also* be displayed, but *with* formatting.

If you want, you can span multiple elements in a text block. E.g.

The diverging pronunciation of tomato (though not so much potato) is primarily one of regional dialect.

The pronunciation 'tuh-MAH-toh' is the standard pronunciation in the UK and is accepted in the US regions of New England along with parts of the lower East Coast, while 'tuh-MAY-toh' is found almost everywhere else.

## Text block limitations

You can’t use text blocks to conditionally display a part of a paragraph. It’s all or nothing. This makes the text block implementation much simpler. However, it might also give some unexpected results:

Also, text blocks in else statements are at the moment not supported by SharpDocx. Instead, use another if statement.

# Loops

You can add repeating text blocks to a document like this:

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **1.**i *squared* is **1**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **2.**i *squared* is **4**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **3.**i *squared* is **9**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **4.**i *squared* is **16**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **5.**i *squared* is **25**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **6.**i *squared* is **36**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **7.**i *squared* is **49**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **8.**i *squared* is **64**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **9.**i *squared* is **81**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **10.**i *squared* is **100**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **11.**i *squared* is **121**

C:\Users\la_113288\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Infobox_info_icon.emfThe value of i is **12.**i *squared* is **144**

Nested loops  
Loops can also be nested.

### Multiples of 1

1 \* 1 = 1

1 \* 2 = 2

1 \* 3 = 3

**Note:** 3 is divisible by 3.

### Multiples of 2

2 \* 1 = 2

2 \* 2 = 4

2 \* 3 = 6

**Note:** 6 is divisible by 3.

### Multiples of 3

3 \* 1 = 3

**Note:** 3 is divisible by 3.

3 \* 2 = 6

**Note:** 6 is divisible by 3.

3 \* 3 = 9

**Note:** 9 is divisible by 3.

### Multiples of 4

4 \* 1 = 4

4 \* 2 = 8

4 \* 3 = 12

**Note:** 12 is divisible by 3.

# Loops, tables and the AppendRow method

Sometimes you do want a loop, but you don’t want a repeating text block. For example, you just want to append rows to a table, but you don’t want to repeat the table itself. In this case, use {! instead of {.

This text and table do NOT repeat, because we used {!. However, a couple of rows do get appended to the table by using the AppendRow method.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i \* 1 | i \* 2 | i \* 3 | i \* 4 | i \* 5 | i \* 6 | i \* 7 | i \* 8 | i \* 9 | i \* 10 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60 |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70 |
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80 |
| 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90 |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| 11 | 22 | 33 | 44 | 55 | 66 | 77 | 88 | 99 | 110 |
| 12 | 24 | 36 | 48 | 60 | 72 | 84 | 96 | 108 | 120 |
| 13 | 26 | 39 | 52 | 65 | 78 | 91 | 104 | 117 | 130 |
| 14 | 28 | 42 | 56 | 70 | 84 | 98 | 112 | 126 | 140 |
| 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 |
| 16 | 32 | 48 | 64 | 80 | 96 | 112 | 128 | 144 | 160 |
| 17 | 34 | 51 | 68 | 85 | 102 | 119 | 136 | 153 | 170 |
| 18 | 36 | 54 | 72 | 90 | 108 | 126 | 144 | 162 | 180 |
| 19 | 38 | 57 | 76 | 95 | 114 | 133 | 152 | 171 | 190 |
| 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 180 | 200 |
| 21 | 42 | 63 | 84 | 105 | 126 | 147 | 168 | 189 | 210 |
| 22 | 44 | 66 | 88 | 110 | 132 | 154 | 176 | 198 | 220 |
| 23 | 46 | 69 | 92 | 115 | 138 | 161 | 184 | 207 | 230 |
| 24 | 48 | 72 | 96 | 120 | 144 | 168 | 192 | 216 | 240 |
| 25 | 50 | 75 | 100 | 125 | 150 | 175 | 200 | 225 | 250 |
| 26 | 52 | 78 | 104 | 130 | 156 | 182 | 208 | 234 | 260 |
| 27 | 54 | 81 | 108 | 135 | 162 | 189 | 216 | 243 | 270 |
| 28 | 56 | 84 | 112 | 140 | 168 | 196 | 224 | 252 | 280 |
| 29 | 58 | 87 | 116 | 145 | 174 | 203 | 232 | 261 | 290 |
| 30 | 60 | 90 | 120 | 150 | 180 | 210 | 240 | 270 | 300 |

## Combining loops, text blocks and tables

You can nest tables in text blocks in order to create multiple tables. Note that the inner loop does not create a repeating text block, but does append rows.

### Multiples of 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i |  | j |  | i \* j |
| 1 | \* | 1 | = | 1 |
| 1 | \* | 2 | = | 2 |
| 1 | \* | 3 | = | 3\* |
| 1 | \* | 4 | = | 4 |
| 1 | \* | 5 | = | 5 |
| 1 | \* | 6 | = | 6\* |
|  |  |  |  | \* Divisible by 3 |

### Multiples of 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i |  | j |  | i \* j |
| 2 | \* | 1 | = | 2 |
| 2 | \* | 2 | = | 4 |
| 2 | \* | 3 | = | 6\* |
| 2 | \* | 4 | = | 8 |
| 2 | \* | 5 | = | 10 |
| 2 | \* | 6 | = | 12\* |
|  |  |  |  | \* Divisible by 3 |

### Multiples of 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i |  | j |  | i \* j |
| 3 | \* | 1 | = | 3\* |
| 3 | \* | 2 | = | 6\* |
| 3 | \* | 3 | = | 9\* |
| 3 | \* | 4 | = | 12\* |
| 3 | \* | 5 | = | 15\* |
| 3 | \* | 6 | = | 18\* |
|  |  |  |  | \* Divisible by 3 |

# Images

Insert images using the Image method.

​

If only a file name is specified, SharpDocx searches this file in a directory specified by the ImageDirectory property. Right now this property has been set to ‘C:\Projects\SharpDocx\Samples\SampleProjects\Tutorial\bin\Debug\net45/../../../../../Images’.

The Image method accepts a second optional parameter that specifies the relative size of the image. Here’s  at 15%.

Images that are too wide to be displayed at 100% are automatically scaled back. Here’s an example:



# Replacing text

If you want to replace text, you can use the Replace method.

This will replace *all* occurrences of the specified string.[[1]](#footnote-2)

Here’s the **replaced text**. And here’s some more replaced text.

# Referencing assemblies and importing namespaces

If you want to use specific types in a view, you can use the Assembly and Import directives to get access to them. Directives look like regular code blocks, but they always start with < %@.

Reference an assembly with the Assembly directive.

Import namespaces with the Import directive.

In C# you would write:

using System.Xml.Linq;

Now we can use types in System.Xml.Linq. Let’s read some news.

**New Fuzzing Tool Finds 26 USB Bugs in Linux, Windows, macOS, and FreeBSD**Academics say they have discovered 26 new vulnerabilities in the USB driver stack employed by operating systems such as Linux, macOs, Windows, and FreeBSD. From a report: The research team, made up by…

**GE Is Saying Goodbye To Its 129-Year-Old Light Bulb Business**schwit1 shares a report from CNN: General Electric is saying goodbye to the light bulb. The conglomerate is shedding a struggling business founded by Thomas Edison more than a century ago. After years…

**Watch SpaceX Launch People To Space For the First Time Live [Updated]**SpaceX is set to mark a huge milestone in its own company history, with a first-ever crewed spaceflight set to take off from Cape Canaveral in Florida later today. From a report: The mission is Commer…

**OpenSSH To Deprecate SHA-1 Logins Due To Security Risk**OpenSSH, the most popular utility for connecting to and managing remote servers, has announced today plans to drop support for its SHA-1 authentication scheme. From a report: The OpenSSH team cited se…

**Huawei CFO Meng Loses Key Court Argument in Fight Against Extradition To United States**Huawei's Chief Financial Officer Meng Wanzhou has lost a key aspect of the trial on her extradition to the United States, a Canadian court announced on Wednesday. From a report: Meng, a Chinese citize…

**Microsoft Releases Windows 10 Update with Linux and Notepad Enhancements**Microsoft is starting to release the latest twice-annual update to Windows 10, featuring enhancements to the longstanding Notepad app and a way to find your cursor in a sea of text. Some of the other …

**Mysterious Radio Bursts Reveal Missing Matter in Cosmos**sciencehabit writes: Roughly half of the "normal" matter in the universe -- the stuff that makes up stars, planets, and even us -- exists as mere wisps of material floating in intergalactic space, acc…

**Fauci: Data is "Really Quite Evident" Against Hydroxychloroquine For Coronavirus**Anthony Fauci told CNN Wednesday that the scientific data "is really quite evident now about the lack of efficacy" of hydroxychloroquine as a coronavirus treatment. From a report: The comments came in…

**Boeing Cutting More Than 12,000 US Jobs With Thousands More Planned**Boeing said Wednesday it was eliminating more than 12,000 U.S. jobs, including laying off 6,770 U.S. workers as the largest American planemaker restructures in the face of the coronavirus pandemic. Fr…

**Local News Stations Run Propaganda Segment Scripted and Produced by Amazon**Local news stations across the U.S. aired a segment produced and scripted by Amazon which touts the company's role in delivering essential groceries and cleaning products during the COVID-19 pandemic,…

**Trump Threatens To Shut Social Media Companies After Twitter Fact Check**President Donald Trump threatened to regulate or shutter social media companies -- a warning apparently aimed at Twitter after it began fact-checking his tweets. From a report: In a pair of tweets iss…

**HBO Max Takes on Netflix With Human Curation Instead of Solely Relying on Algorithms**Just like nearly everything else on the internet, streaming services are ruled by recommendation algorithms that are designed to predetermine what people want before they ask for it. WarnerMedia is tr…

**The CDC Says Its New 'Best Estimate' Is That 0.4 Percent of People With Symptoms and COVID-19 Will Die**An anonymous reader quotes a report from CNN: In new guidance for mathematical modelers and public health officials, the U.S. Centers for Disease Control and Prevention is estimating that about a thir…

**CDC Warns of Increasingly Aggressive Rodents Looking For New Food Sources**New submitter Way Smarter Than You shares a report from The Seattle Times: Humans aren't the only ones hankering for the days they could dine out at their cities' restaurants: Some rats that miss feas…

**Insignia Project Aims To Resurrect Xbox Live For the Original Xbox**Last week, Kotaku reported on a new project, called Insignia, "that aims to recreate the original Xbox Live service, potentially restoring online play to many dozens of classic Xbox games that fell of…

In a real world scenario you wouldn’t fetch data or have this much code in a view. But hey, this is just an example.

# The Map

The Map maps OpenXmlElements to plain text and vice versa. It’s being used internally by the Replace method and for finding the C# code in views, among other things. At the moment Map.Text looks something like this:

Version 2.0.2.0  
egonl  
May 2020  
Version 2.0.2.0  
egonl  
May 2020  
SharpDocx  
SharpDocx  
  
  
Summary  
Generating documents with SharpDocx is a two-step process. First you create a view in Word. A view is a Word document which also contains C# code. Code can be inserted anywhere, e.g. 28-5-2020 00:09:20 would insert the current date and time.  
The next step is to create documents based on this view. This requires two lines of code:  
 var document = DocumentFactory.Create("view.cs.docx");  
 document.Gene …

The Map might be handy when you want to search the document for text.

# The SharpDocx solution

### Building the example programs

The Tutorial, Inheritance and Model samples will by default be build for .NET Framework 3.5, 4.5 and .NET Core 2.0. The .NET Core 2.0 builds will use the .NET Standard 2.0 version of SharpDocx.

The Tutorial and Inheritance samples will by default run in .NET Framework 4.5. The Model sample will by default run in .NET Core 2.0. If you want to change this, right click on the solution file in Visual Studio and select **Edit SampleName.csproj**. This will open the csproj-file. The first target named on this line will be used for startup/debugging in Visual Studio:

<TargetFrameworks>net45;net35;netcoreapp2.0</TargetFrameworks

### Unix-like systems

On Unix-like systems, remove the net45 and net35 targets from all projects, because they are only available on Windows. Other than that, the library and samples should compile and run fine.

1. Actually, this will only replace text in the body of the document, and not in headers, footers, end- or footnotes. So this won’t work as expected. But you can use code here. [↑](#footnote-ref-2)