

# How can a Word document be created in C#? [closed]

Asked 16 years, 4 months ago   Modified 5 years, 1 month ago

Viewed 83k times



74



**Closed.** This question needs to be more [focused](#). It is not currently accepting answers.



**Want to improve this question?** Update the question so it focuses on one problem only by [editing this post](#).

Closed 11 years ago.

[Improve this question](#)

I have a project where I would like to generate a report export in MS Word format. The report will include images/graphs, tables, and text. What is the best way to do this? Third party tools? What are your experiences?

c#

.net

ms-word

openxml

Share

Improve this question

Follow

edited Feb 3, 2010 at 20:32



Peter Mortensen

31.6k ● 22 ● 109 ● 133

asked Aug 13, 2008 at 22:07



Schmidty

1,899 ● 1 ● 20 ● 18

- 
- 6 This should be re-opened. Not only is it clear and useful, but is linked to from other questions which are marked as duplicate of this. – [DarenW](#) Aug 17, 2016 at 7:19
- 

17 Answers

Sorted by:

Highest score (default)



49



The answer is going to depend slightly upon if the application is running on a server or if it is running on the client machine. If you are running on a server then you are going to want to use one of the XML based office generation formats as there are know issues when [using Office Automation on a server](#).

However, if you are working on the client machine then you have a choice of either [using Office Automation](#) or using the Office Open XML format (see links below), which is supported by Microsoft Office 2000 and up either natively or through service packs. One draw back to this though is that you might not be able to embed some kinds of graphs or images that you wish to show.

The best way to go about things will all depend sightly upon how much time you have to invest in development. If you go the route of Office Automation there are quite a few good tutorials out there that can be found via Google and is fairly simple to learn. However, the Open Office

XML format is fairly new so you might find the learning curve to be a bit higher.

## Office Open XML Information

- Office Open XML -  
[http://en.wikipedia.org/wiki/Office\\_Open\\_XML](http://en.wikipedia.org/wiki/Office_Open_XML)
- OpenXML Developer -  
<http://openxmldeveloper.org/default.aspx>
- Introducing the Office (2007) Open XML File Formats  
- <http://msdn.microsoft.com/en-us/library/aa338205.aspx>

Share Improve this answer

edited Aug 25, 2009 at 5:23

Follow

answered Aug 13, 2008 at 22:19



rjzii

14.5k ● 12 ● 81 ● 122

- 
- 6 Update as of 9/2011...I do both Interop and OpenXml, and faced with this task in either case, I would absolutely use OpenXml. It's the only solution which absolutely guarantees you'll have absolute control over the output.
- [Chris B. Behrens](#) Sep 13, 2011 at 14:30
- 



17

[DocX free library](#) for creating DocX documents, actively developed and very easy and intuitive to use. Since CodePlex is dying, project has moved to [github](#).



Share Improve this answer

edited Apr 27, 2017 at 15:49

Follow



answered Jul 22, 2010 at 15:07



**MadBoy**

11.1k ● 24 ● 102 ● 163



14



I have spent the last week or so getting up to speed on [Office Open XML](#). We have a database application that stores survey data that we want to report in Microsoft Word. You can actually create Word 2007 (docx) files from scratch in C#. The Open XML SDK version 2 includes a cool application called the Document Reflector that will actually provide the C# code to fully recreate a Word document. You can use parts or all of the code, and substitute the bits you want to change on the fly. The help file included with the SDK has some good code samples as well.

There is no need for the Office Interop or any other Office software on the server - the new formats are 100% XML.

Share Improve this answer

edited Feb 3, 2010 at 20:38

Follow



**Peter Mortensen**

31.6k ● 22 ● 109 ● 133

answered May 20, 2009 at 13:40



**Eric Flamm**



10



Have you considered using .RTF as an alternative?

It supports embedding images and tables as well as text, opens by default using Microsoft Word and whilst it's featureset is more limited (count out any advanced formatting) for something that looks and feels and opens like a Word document it's not far off.

Your end users probably won't notice.

Share Improve this answer

answered Aug 14, 2008 at 14:43

Follow



[DavidWhitney](#)

4,378 ● 4 ● 25 ● 30

---

1 Please don't use RTF - the spec is a mess, even Microsoft does not follow it fully, and it has lots of ambiguity in it.

– [David Thielen](#) Jun 6, 2012 at 22:17

---



10



I have found [Aspose Words](#) to be the best as not everybody can open Office Open XML/\*.docx format files and the Word interop and Word automation can be buggy. Aspose Words supports most document file types from Word 97 upwards.

It is a pay-for component but has great support. The other alternative as already suggested is RTF.

Share Improve this answer

edited Feb 3, 2010 at 20:42

Follow



[Peter Mortensen](#)

31.6k ● 22 ● 109 ● 133

answered Aug 14, 2008 at 14:57



John

30.5k ● 18 ● 92 ● 133

---

I've found Aspose Words to be a bit lacking when working with OpenXml documents. Specifically, its handling of content controls is next to useless, and it cannot handle AltChunk nodes at all. – Collin K Oct 25, 2011 at 20:36

---



8



To generate Word documents with Office Automation within .NET, specifically in C# or VB.NET:

1. Add the **Microsoft.Office.Interop.Word** assembly reference to your project. The path is **Visual Studio Tools for Office\PIA\Office11\Microsoft.Office.Interop.Word.dll**.
2. Follow the Microsoft code example you can find here: <http://support.microsoft.com/kb/316384/en-us>.

Share Improve this answer

Follow

edited Feb 3, 2010 at 20:34



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Dec 1, 2009 at 22:50



Marcello Belguardi

101 ● 2 ● 3



Schmidty, if you want to generate Word documents on a web server you will need a licence for each **client** (not

4

just the web server). See this section in the first [link](#) Rob posted:



"Besides the technical problems, you must also consider licensing issues. Current licensing guidelines prevent Office applications from being used on a server to service client requests, unless those clients themselves have licensed copies of Office. Using server-side Automation to provide Office functionality to unlicensed workstations is not covered by the End User License Agreement (EULA)."

If you meet the licensing requirements, I think you will need to use COM Interop - to be specific, the [Office XP Primary Interop Assemblies](#).

Share Improve this answer

answered Aug 13, 2008 at 23:21

Follow



Luke Girvin

13.4k ● 10 ● 67 ● 85



3



Check out VSTO (Visual Studio Tools for Office). It is fairly simple to create a Word template, inject an xml data island into it, then send it to the client. When the user opens the doc in Word, Word reads the xml and transforms it into WordML and renders it. You will want to look at the ServerDocument class of the VSTO library. No extra licensing is required from my experience.



3



I have had good success using the Syncfusion Backoffice DocIO which supports doc and docx formats.

In prior releases it did not support everything in word, but according to your list we tested it with tables and text as a mail merge approach and it worked fine.



Not sure about the import of images though. On their blurb page

<http://www.syncfusion.com/products/DocIO/Backoffice/features/default.aspx> it says

Blockquote Essential DocIO has support for inserting both Scalar and Vector images into the document, in almost all formats. Bitmap, gif, png and tiff are some of the common image types supported.

So its worth considering.

As others have mentioned you can build up a RTF document, there are some good RTF libraries around for .net like

<http://www.codeproject.com/KB/string/nrtftree.aspx>



Follow



Jafin

4,351 ● 1 ● 42 ● 54



3



I faced this problem and created a small library for this. It was used in several projects and then I decided to publish it. It is free and very very simple but I'm sure it will help with you with the task. Invoke the Office Open XML Library, <http://invoke.co.nz/products/docx.aspx>.



Share Improve this answer

edited Feb 3, 2010 at 20:39



Follow



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Dec 25, 2008 at 21:51



Scott



3



I've written a [blog post series on Open XML WordprocessingML document generation](#). My approach is that you create a template document that contains content controls, and in each content control you write an XPath expression that defines how to retrieve the content from an XML document that contains the data that drives the document generation process. The code is free, and is licensed under the the [Microsoft Reciprocal License \(Ms-RL\)](#). In that same blog post series, I also explore an approach where you write C# code in content controls. The document generation process then processes the template document and generates a C# program that generates the desired documents. One advantage of this



approach is that you can use *any* data source as the source of data for the document generation process. That code is also licenced under the Microsoft Reciprocal License.

Share Improve this answer

answered May 12, 2011 at 3:50

Follow



Eric White

1,881 ● 12 ● 15



2



I currently do this exact thing.

If the document isn't very big, doesn't contain images and such, then I store it as an RTF with `#MergeFields#` in it and simply replace them with content, sending the result down to the user as an RTF.



For larger documents, including images and dynamically inserted images, I save the initial Word document as a Single Webpage \*.mht file containing the `#MergeFields#` again. I then do the same as above. Using this, I can easily render a DataTable with some basic Html table tags and replace one of the `#MergeFields#` with a whole table.

Images can be stored on your server and the url embedded into the document too.

Interestingly, the new Office 2007 file formats are actually zip files - if you rename the extension to .zip you can open them up and see their contents. This means you

should be able to switch content such as images in and out using a simple C# zip library.

Share Improve this answer

answered Sep 15, 2008 at 11:05

Follow



littlecharva

4,244 ● 9 ● 47 ● 54



1

@Dale Ragan: That will work for the Office 2003 XML format, but that's not portable (as, say, .doc or .docx files would be).



To read/write those, you'll need to use the Word Object Library ActiveX control:



<http://www.codeproject.com/KB/aspnet/wordapplication.aspx>

Share Improve this answer

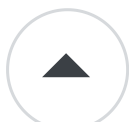
answered Aug 13, 2008 at 22:21

Follow



TheSmurf

15.6k ● 3 ● 41 ● 48



1

@Danny Smurf: Actually this article describes what will become the Office Open XML format which Rob answered with. I will pay more attention to the links I post for now on to make sure there not obsolete. I actually did a search on WordML, which is what it was called at the time.

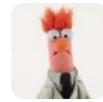


I believe that the Office Open XML format is the best way to go.

Share Improve this answer

answered Aug 13, 2008 at 22:34

Follow



**Dale Ragan**

18.3k ● 3 ● 55 ● 71



1

[LibreOffice](#) also supports headless interaction via API. Unfortunately there's currently not much information about this feature yet.. :(



Share Improve this answer

answered Nov 14, 2011 at 9:56

Follow



**plaes**

32.6k ● 11 ● 94 ● 89



1

You could also use Word document generator. It can be used for client-side or server-side deployment. From the project description:



WordDocumentGenerator is an utility to generate Word documents from templates using Visual Studio 2010 and Open XML 2.0 SDK.

WordDocumentGenerator helps generate Word documents both non-refresh-able as well as refresh-able based on predefined templates using minimum code changes. Content controls are used as placeholders for document generation. It supports Word 2007 and Word 2010.

Grab it: <http://worddocgenerator.codeplex.com/>

Download SDK: <http://www.microsoft.com/en-us/download/details.aspx?id=5124>

Share Improve this answer

answered Dec 27, 2012 at 19:08

Follow



Ricardo

555 ● 5 ● 9



0



Another alternative is [Windward Docgen](#) (disclaimer - I'm the founder). With Windward you design the template in Word, including images, tables, graphs, gauges, and anything else you want. You can set tags where data from an XML or SQL datasource is inserted (including functionality like forEach loops, import, etc). And then generate the report to DOCX, PDF, HTML, etc.

Share Improve this answer

edited Nov 9, 2019 at 21:21

Follow

answered Jun 6, 2012 at 22:37



David Thielen

32.7k ● 40 ● 144 ● 263

2 FYI, when I go to a site for a commercial product and it seems that they hide the price while touting that there is a free trial I automatically assume the product is really expensive and the free trial is just bait. If Docgen is crazy expensive then you're probably right to hide the price but otherwise I think it's a mistake. – [Steve Hiner](#) Jan 27, 2017 at 15:38

@SteveHiner - We have price ranges there (search on price) but we took down the price list because a significant percentage of the time people misunderstood what they needed and then felt misled. So we figure better to understand someone's need and then give them the right price. – [David Thielen](#) Jan 28, 2017 at 23:24

---

- 1 @SteveHiner - You were right. After a lot of people complained we put the prices up on the website, and are trying to explain it well enough they understand what it will be for them. – [David Thielen](#) Feb 8, 2019 at 21:05
- 



**Highly active question.** Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.