

RSS Feed for my Blog

Search  Go

Articles

- 2014 (9)
- 2013 (12)
- 2012 (12)
- 2011 (12)
- 2010 (12)
- 2009 (12)
- 2008 (12)
- 2007 (12)
- 2006 (12)
- 2005 (12)
- 2004 (12)
- 2003 (12)

Blog Archives

- October 2014 (1)
- September 2014 (1)
- August 2014 (1)
- July 2014 (3)
- June 2014 (3)
- May 2014 (4)
- April 2014 (8)
- March 2014 (6)
- February 2014 (1)
- January 2014 (3)
- December 2013 (1)
- November 2013 (3)
- October 2013 (3)
- September 2013 (2)
- August 2013 (1)
- July 2013 (4)
- June 2013 (4)
- May 2013 (2)
- April 2013 (3)
- March 2013 (7)
- February 2013 (3)
- January 2013 (3)
- December 2012 (7)
- November 2012 (6)
- October 2012 (5)
- September 2012 (4)
- August 2012 (5)
- July 2012 (5)
- June 2012 (4)
- May 2012 (2)
- April 2012 (3)
- March 2012 (7)
- February 2012 (5)
- January 2012 (5)
- December 2011 (3)
- November 2011 (3)
- October 2011 (5)
- September 2011 (3)
- August 2011 (4)
- July 2011 (5)
- June 2011 (3)
- May 2011 (2)
- April 2011 (4)
- March 2011 (2)
- February 2011 (5)
- January 2011 (10)
- December 2010 (6)
- November 2010 (8)
- October 2010 (5)
- September 2010 (7)
- August 2010 (4)
- July 2010 (4)
- June 2010 (3)
- May 2010 (4)

## Using OpenXML to fill a Word document from a .Net application

Thursday, March 24, 2011

Print this page

Download source code

Everybody knows that using Office automation from an application can be a real pain. Even Microsoft itself does not recommend it for server applicati (<http://support.microsoft.com/kb/257757>).

So what are the alternatives? I am personally a big fan of [Aspose components](#). But if you cannot afford it, there are some other ways of dealing with t documents (now that they moved to a XML format). What I will explore here is a simple method to fill Word documents (version 2007 and later) usin SDK. This SDK has nothing to do with automation and can be used on a PC/server that does not have Office installed at all.

### Demo Code

This month demo code is available in both VB and C#. It has been created using Visual Studio 2008.

### What you need

You will first need to download and install the "Open XML SDK 2.0 for Microsoft Office" from <http://www.microsoft.com/downloads/en/details.aspx?F36E9-45F5-8D8C-331DF206E0D0&displaylang=en>. In order to be able to run this SDK, you must also at least use Visual Studio 2008 SP1 (and target 3.5).

Users who will open the filled document will need either Office 2007 or later or an older version with the Compatibility pack (<http://www.microsoft.com/downloads/en/details.aspx?familyid=941b3470-3ae9-4aee-8f43-c6bb74cd1466&displaylang=en>) properly installed.

### Creating a template document

The scenario here is a simple one where we will fill an existing .docx file. That means that this document must already exist. The great thing about a t at run time is that users can easily modify the content and the formatting without affecting the results (as long as our fields still exist).

Your template must contain a number of MERGEFIELD placeholders. Later, we will find these placeholders to pass values to them.

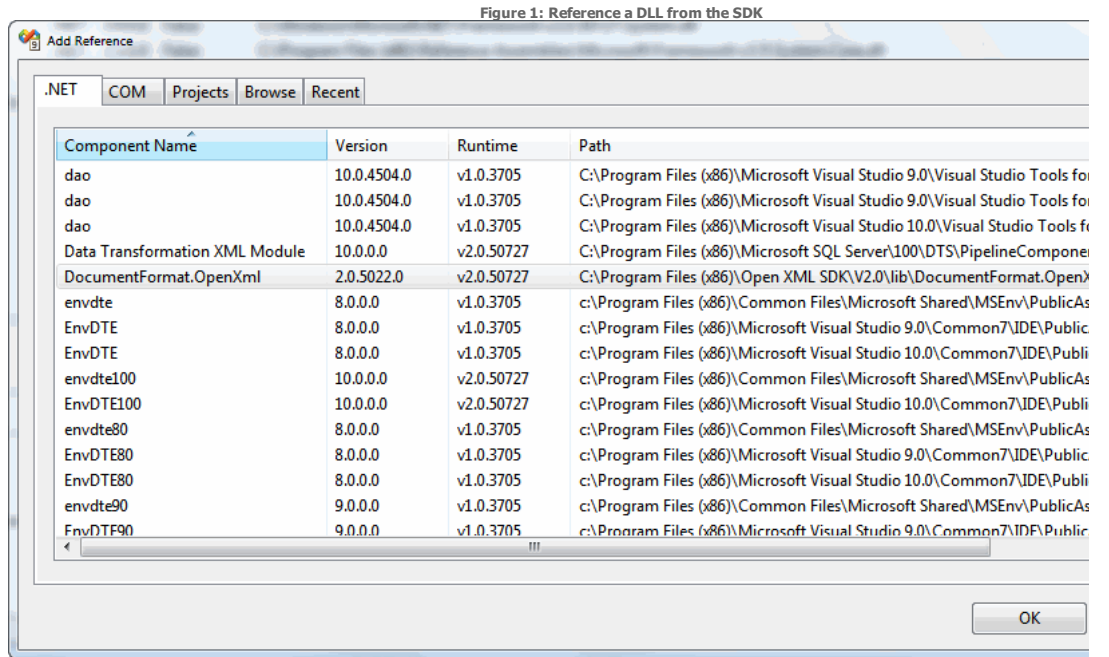
The biggest benefit of using templates is that a user could provide a template or modify it and your application will still be able to fill as long as the u MERGEFIELD placeholders. Because the merge fields are named, you can move them around your documents.

I have created a small template (named Template.docx) using Word 2010. This template contains 3 fields: Date, Time, and FreeText.

This template has been added to the solution and I have set the "Copy to Output Directory" to "Copy always" so that it always gets copied to the bin | solution is compiled.

### Creating the project

When you create your project, you need to add a reference to a DLL you installed from the SDK. This DLL is DocumentFormat.OpenXML.dll as shown |



The demo application contains a single form. This form lets you specify a template (Template.docx by default) and the other textbox let you enter a t pass to the FreeText merge field. Finally, the "Fill Document" button does the entire job (notice that you cannot run the job twice because the templa merge fields anymore, restarting the demo recopies the template).

Figure 2: The demo application running

[April 2010 \(5\)](#)  
[March 2010 \(5\)](#)  
[February 2010 \(3\)](#)  
[January 2010 \(4\)](#)  
[December 2009 \(4\)](#)  
[November 2009 \(3\)](#)  
[October 2009 \(7\)](#)  
[September 2009 \(5\)](#)  
[August 2009 \(4\)](#)  
[July 2009 \(5\)](#)  
[June 2009 \(7\)](#)  
[May 2009 \(9\)](#)  
[April 2009 \(7\)](#)  
[March 2009 \(10\)](#)  
[February 2009 \(5\)](#)  
[January 2009 \(12\)](#)  
[December 2008 \(9\)](#)  
[November 2008 \(9\)](#)  
[October 2008 \(15\)](#)  
[September 2008 \(9\)](#)  
[August 2008 \(15\)](#)  
[July 2008 \(5\)](#)  
[June 2008 \(8\)](#)  
[May 2008 \(7\)](#)  
[April 2008 \(9\)](#)  
[March 2008 \(13\)](#)  
[February 2008 \(7\)](#)  
[January 2008 \(5\)](#)  
[December 2007 \(5\)](#)  
[November 2007 \(6\)](#)  
[October 2007 \(6\)](#)  
[September 2007 \(3\)](#)  
[August 2007 \(3\)](#)  
[July 2007 \(6\)](#)  
[June 2007 \(10\)](#)  
[May 2007 \(7\)](#)  
[April 2007 \(6\)](#)  
[March 2007 \(7\)](#)  
[February 2007 \(5\)](#)  
[January 2007 \(3\)](#)

## MSDEVMTL

EST. Jul 08, 2011



1,126 Membres

Oct 28, 2014 | 4:00 AM

[Groupe architecture: Microsoft Dynamics CRM, vue d'ensemble de la plateforme](#)

Hyatt Regency Montréal (Salle Symphonie 3) - Montréal, QC

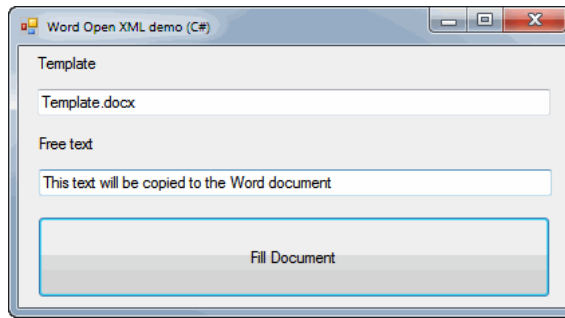
[JOIN](#)

[ADD THIS TO YOUR SITE](#)

Group Rating

### My favorite links

[Communauté des développeurs](#)  
[Microsoft de Montréal](#)  
[DevTeach conference](#)  
[TechDays.ca conference](#)  
[ScottGu's Blog](#)  
[MSDN ASP.NET](#)  
[Experts Exchange](#)



Here is a breakdown of the important parts of code.

### Opening the template

Opening the Word document file is very easy with the SDK. Only the following code is required:

```
Dim objWordDocx As WordprocessingDocument
objWordDocx = WordprocessingDocument.Open(txtTemplate.Text, True)
```

Once you have the document open, you need to get a pointer to the part of the document you want to work with. Here, we will be using the main doc could also access the header part and/or the footer part for example):

```
Dim objMainDoc As OpenXmlElement
objMainDoc = objWordDocx.MainDocumentPart.Document()
```

### Looping through the fields

Here is the meal of this technique. You need to find the fields in the template and replace them with some values you have in your application.

The first thing you need to do is to loop through the fields that are declared in your template:

```
For Each objField As SimpleField In objMainDoc.Descendants(Of SimpleField) ()
    ...
Next
```

Inside that loop, you get a reference to a SimpleField object from which you need to extract the field name (GetFieldName is available in the download method to extract the field name only):

```
'Clean the field name
Dim strFieldName As String = GetFieldName(objField)
```

Now that you have the field name, you can check if it is valid and if you need to process it (in my case, I check if the field exist in dictionary of fields know we have a field to process, we have to extract the XML (using the RunProperties object) and append new text to it:

```
If Not String.IsNullOrEmpty(strFieldName) Then
    'check if we have a value for this merge field
    If values.ContainsKey(strFieldName) AndAlso Not String.IsNullOrEmpty(values(strFieldName)) Then
        'Find the XML placeholder
        Dim objRunProp As String = String.Empty
        For Each objRP As RunProperties In objField.Descendants(Of RunProperties) ()
            objRunProp = objRP.OuterXml
            Exit For
        Next
        Dim objRun As New Run
        If Not String.IsNullOrEmpty(objRunProp) Then
            objRun.Append(New RunProperties(objRunProp))
        End If
        'add the text to the place holder
        objRun.Append(New Text(values(strFieldName)))

        'replace the merge field with the value
        objField.Parent.ReplaceChild(Of SimpleField)(objRun, objField)
    End If
End If
```

### Saving and closing the document

The last thing you need to do is to save the document. First, I save the part of the document I worked with (could also be the header or the footer), document itself:

```
objWordDocx.MainDocumentPart.Document.Save()
objWordDocx.Close()
```

### Conclusion

Using the OpenXML component will be enough to satisfy many needs of filling Office documents (Word in this case here) without having to install the a server or PC. This also means that you can save the price of this license!



Download source code

Share this post: [Email it!](#)

Posted by Eric

### Comments

**re: Using OpenXML to fill a Word document from a .Net application**

Tuesday, August 02, 2011 8:09 AM by Eric Moreau

**Yeah We're Geeks**  
when it comes to web hosting. It makes us better.

Windows Server 2008 R2 /  
IIS 7.5 / ASP.NET 3.5 /  
ASP.NET 4 / SQL Server 2008 /  
Silverlight / ASP.NET MVC

**orcsw**  
Managed Hosting Solutions



Interesting use of OpenXML to parse an Excel file: <http://openxmldeveloper.org/blog/b/openxmldeveloper/archive/2011/07/18/query-open-xml-spreadsheet.aspx>

 **re: Using OpenXML to fill a Word document from a .Net application**

Thursday, November 17, 2011 8:03 AM by taz

hi. this page was very useful for a newbie like me. much appreciated, many thanks!

 **re: Using OpenXML to fill a Word document from a .Net application**

Thursday, February 02, 2012 5:20 PM by Rony Olazabal

Because when you change or add more text to the template does not show the fields in Microsoft Word 2010 document?

 **re: Using OpenXML to fill a Word document from a .Net application**

Sunday, May 06, 2012 9:03 PM by Eric Moreau

check the sample from <http://www.codeproject.com/Articles/38575/Fill-Mergefields-in-docx-Documents-without-Microso>

 **re: Using OpenXML to fill a Word document from a .Net application**

Monday, December 03, 2012 1:47 AM by Gurudatt

We have developed an app which is working good based on this article. But recently we have had requirements to introduce conditional statements for the template, something like `{If {MERGEFIELD netToBorrower \* MERGEFORMAT } <> 0 "Incentive" "Zero"}` However hard we try, we are unable to get the Se the conditional statement to work. Also if there is a formula to be applied something like `{SET calculatedPayoffAmount {={Mergefield payoffAmount } + { fundsPostedToAccount }}}}` However I try, I am unable to get this working. Do we have a solution for the above 2 questions? Kindly let me know if there is working

 **re: Using OpenXML to fill a Word document from a .Net application**

Monday, December 03, 2012 6:09 AM by Eric Moreau

I never used conditions in my mail merge. I don't know if it is only possible. But I am pretty sure you can ensure that the data you merge already contains Your condition should really be in your code that is preparing your data.

 **re: Using OpenXML to fill a Word document from a .Net application**

Thursday, December 20, 2012 8:49 AM by David

I am exploring ways of generating docx files dynamically in .NET. This was a very helpful and complete example. Thanks for taking the time to share!

 **re: Using OpenXML to fill a Word document from a .Net application**

Tuesday, February 12, 2013 12:23 AM by Janos Nagy

Thanks for this demo and explanations. I was looking for just this kind of get starting tutorial. I have tried to develop the example but unfortunately probl understand. The problem is in the word-document I use as a template. I will start with your document and try to adjust it for my purposes. As soon as I add to the document it becomes corrupted somehow and the line below For Each oFLD In oDoc.MainDocumentPart.Document.Descendants(Of SimpleField)() r the old ones from you and the new one. I tried with a completely new document and I noticed that when I put the fields in one and the same sequence th works and I can find them by `oDoc.MainDocumentPart.Document.Descendants(Of SimpleField)()` but when I save the document in Word between fields th finds no SimpleField in `oDoc.MainDocumentPart.Document.Descendants`. Any idea what this is about, please?

 **re: Using OpenXML to fill a Word document from a .Net application**

Tuesday, February 12, 2013 8:28 AM by Eric Moreau

Never had problem with that kind of corrupted file. Maybe related to the version of Word!

 **re: Using OpenXML to fill a Word document from a .Net application**

Saturday, February 16, 2013 6:00 PM by Jake

Hi Eric, Thank you. It is very helpful. Save me many hours of searching for solutions. Thanks again.

 **re: Using OpenXML to fill a Word document from a .Net application**

Tuesday, February 26, 2013 7:41 AM by Thomas

Hello, I'm having the exact same problem as Janos : Word seems to save mergefields in a completely random way. I developed a little app in order to list a a given document. When I add my first mergefield in the document, the app finds it. If I add a second mergefield, the app can only find the second field. mergefield, the app finds the first and third field, and ignores the second one ... If I add more mergefields, the app randomly finds some fields, and ignore Upon inspecting the XML document, it appears that the mergefields are not always saved as a "fldSimple" XML element, which explains why we can't find e the document with the above code. Can anyone explain why Word won't save each mergefield in the document as a SimpleField ? Is this an expected behi ? Thanks in advance

 **re: Using OpenXML to fill a Word document from a .Net application**

Sunday, November 03, 2013 7:23 AM by Alex Kahn

Very helpful article! But some questions arise when reading it. If I want to save a document under the other name, what should I do? Then, how to add to document the pagebreak and to add the new content, where placeholders of mergefields are replaced with the data from any table? (This table is run-time templates is used many times not one. Thanx in advance

 **re: Using OpenXML to fill a Word document from a .Net application**

Sunday, November 03, 2013 7:49 PM by Eric Moreau

If you want to save under a different name, you better copy the original document then open the copy and save it. This way, your original remains untouched, check <http://stackoverflow.com/questions/2795315/create-page-break-using-openxml>.

 **re: Using OpenXML to fill a Word document from a .Net application**

Monday, November 04, 2013 5:54 AM by Alex Kahn

Thank you for the very quick reply. With pagebreaks I understood, but how I should insert the text of the template (with replaced mergefields from record table) many times (to be equal to my table record count) to the Word document?

 **re: Using OpenXML to fill a Word document from a .Net application**

Monday, November 04, 2013 6:43 AM by Eric Moreau

Check [http://msdn.microsoft.com/en-us/library/office/bb508029\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/office/bb508029(v=office.12).aspx)

 **re: Using OpenXML to fill a Word document from a .Net application**

Monday, November 04, 2013 10:56 AM by Alex Kahn

Thank you for the very quick reply. With pagebreaks I understood, but how I should insert the text of the template (with replaced mergefields from record table) many times (to be equal to my table record count) to the Word document?

 **re: Using OpenXML to fill a Word document from a .Net application**

Tuesday, April 22, 2014 3:49 AM by Ricky Simanjuntak

How to modified or create new file? because if I modified the sample file (Template.docx) the application will not replace the mergefield with the new value advance.

 **re: Using OpenXML to fill a Word document from a .Net application**

Tuesday, April 22, 2014 8:21 AM by Eric Moreau

I just re-downloaded the demo application and it still works replacing all my merge fields. I have also added a new field and this last one got updated properly. I know what happened to your template.

**Leave a comment**

Author:

E-mail:  (Optional)

Comment:



Enter the code shown: