Are Word 2013 repeatingSection content controls ready for prime time?

# Summary (TL;DR)

Word 2013 introduced a native repeating content control, which could potentially be used instead of an OpenDoPE one.

However, Word 2010 strips out Word 2013 repeating content controls, and does so without warning! Compare OpenDoPE repeats, which work in Word 2007, 2010 and 2013.

So until Word 2010 becomes irrelevant (or support is back ported), Word 2013 repeating content controls can’t be used in a generic solution.

In addition, Word 2013 doesn’t handle the case of repeat zero times as you’d expect; it leaves a single instance, which will cause problems in many applications.

Even so, docx4j 3.2.2 will support processing Word 2013 repeating content controls, for those who still choose to use them.

# Introduction

For developers wondering about Microsoft’s commitment to content controls, Office 2013 was certainly good news.

In Microsoft’s “What's new for Word 2013 developers”, 2 of the 4 items were about content controls:

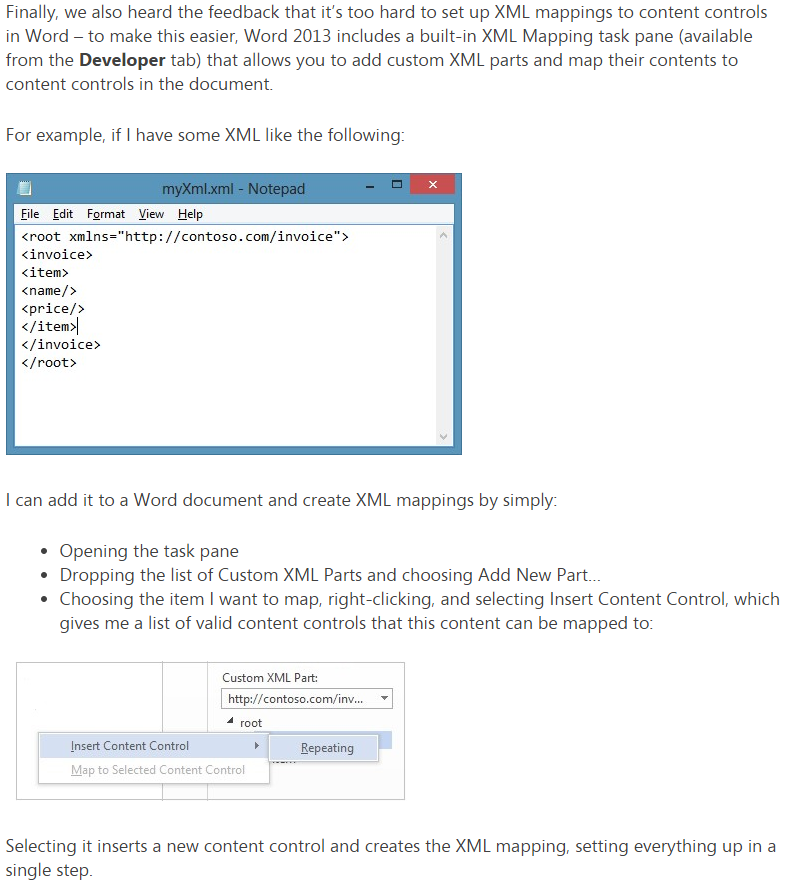
* Enhancements to content controls
* UI for XML mappings

And in <http://blogs.office.com/2012/10/17/top-5-reasons-developers-will-love-word-2013/> 4 of the 5 reasons relate to content controls!

The MSDN article “What's new with content controls in Word 2013” describes the changes in detail; here we’ll focus on repeating section content controls, and whether the time is ripe to migrate from OpenDoPE repeats to these.

# XML mapping pane

As top-5-reasons-developers-will-love-word-2013 explains:



The XML mapping pane is certainly a welcome addition. (It is based on – or at least similar in functionality to <http://xmlmapping.codeplex.com/> which a more recent OpenDoPE add in borrows from)

However, mapping repeats with the XML mapping pane can be confusing, since the tool might not give much feedback.

We’ll use the following simple XML (for an invoice) so you can see what I mean:

<inv:invoice xmlns:inv="http://contoso.com/invoice">

<inv:logo>iVBORw0KGgoAAAA...base64 encoded...lFTkSuQmCC</inv:logo>

<inv:invoicenumber>1234</inv:invoicenumber>

<inv:invoicedate>DATE</inv:invoicedate>

<inv:customer>

<inv:contact>John Citizen</inv:contact>

<inv:company>Contoso Inc</inv:company>

<inv:address>

<inv:addressline1>Unit 4</inv:addressline1>

<inv:addressline2>some street</inv:addressline2>

<inv:suburb>Darling Point</inv:suburb>

<inv:state>NSW</inv:state>

<inv:postcode>2027</inv:postcode>

</inv:address>

</inv:customer>

<inv:purchaseorder>

<inv:number>PO1234</inv:number>

</inv:purchaseorder>

<inv:lines>

<inv:lineitem>

<inv:productcode>ITEM1</inv:productcode>

<inv:description>Some item</inv:description>

<inv:quantity>2</inv:quantity>

<inv:price>$120</inv:price>

</inv:lineitem>

<inv:lineitem>

<inv:productcode>ITEM2</inv:productcode>

<inv:description>Another item</inv:description>

<inv:quantity>3</inv:quantity>

<inv:price>$90</inv:price>

</inv:lineitem>

</inv:lines>

<inv:VAT applies="true" />

<inv:terms>&lt;?xml version="1.0" standalone="yes"?&gt;&lt;?mso-application progid="Word.Document"?&gt;&lt;pkg:package xmlns:pkg="http://schemas.microsoft.com/office/2006/xmlPackage"&gt;

:

:

&lt;/pkg:package&gt;

</inv:terms>

<inv:notes>

<inv:note>note 1</inv:note>

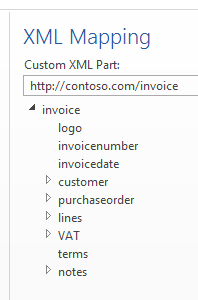
<inv:note>note 2</inv:note>

</inv:notes>

</inv:invoice>

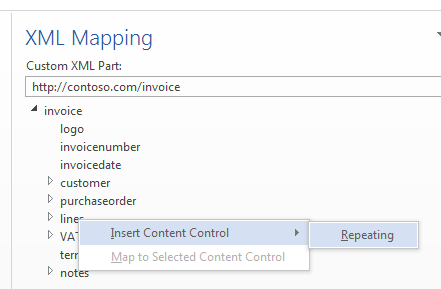
If you have Word 2013 and want to follow along, you can create a new document, go to the DEVELOPER tab, click “XML Mapping Pane”, then “add new part”, and add the above XML.

Now you can select the invoice part, and you should see:

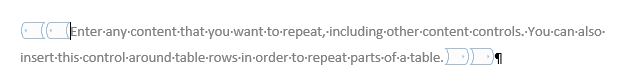


## Tip #1: The first tip is to make sure you aren’t in Design Mode!

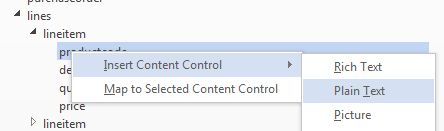
Suppose you accept the suggestion to map “lines” to a repeat:



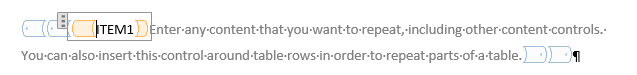
When you do that, you should see on the document surface:



OK, now map lines/lineitem[1]/product code:

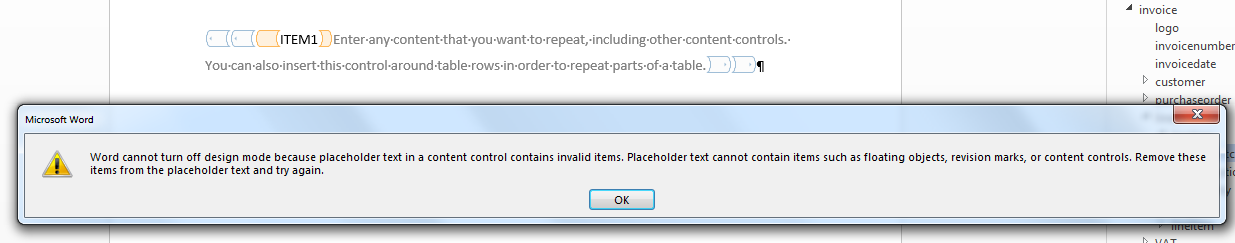


You get:



And you could keep mapping other items, thinking everything is OK.

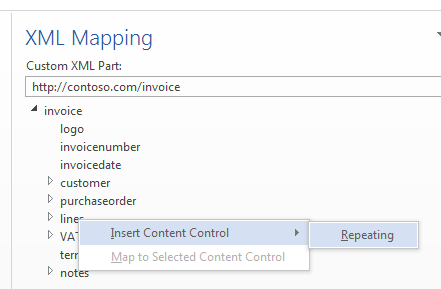
But as soon as you try to toggle design mode off, you get a nasty message:



(though Word does let you save & close the docx, and when you re-open it, you’re out of design mode … but you’ve still set things up wrong).

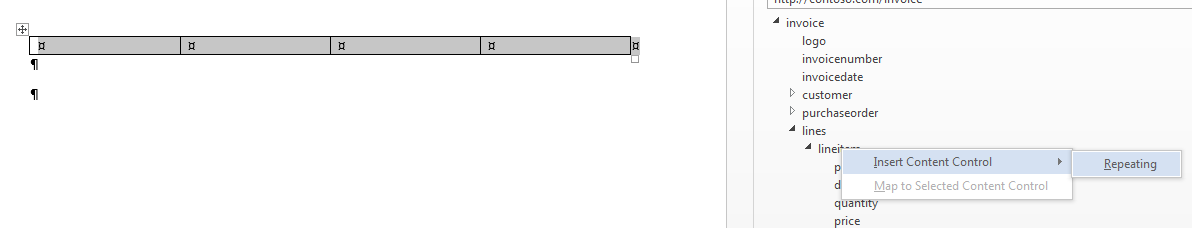
## Tip #2: be very wary about the tools suggestions as to what to map.

As shown in the image above, Word leads you down the path of mapping lines to a repeat:



Actually, that’s not what we want. We need to map lines/lineitem! To create a repeat, you should be mapping the first child in the list of items being repeated, not the container.

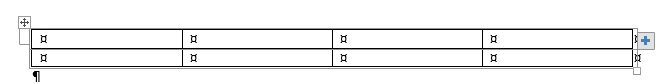
Let’s map that to a table row (4 columns, 1 per child). To do that, first we select the table row:



Note: in some cases, the mapping pane won’t even let you map your repeat properly. Case in point: the notes element in this example, where you need to map the first note to a repeat, but aren’t given that as an option.

## Tip #3: don’t be put off by incomplete visual feedback

Word gives you some indication that you’ve done the repeat correctly:



First, the “+” sign is there.

And second, you now have 2 rows.

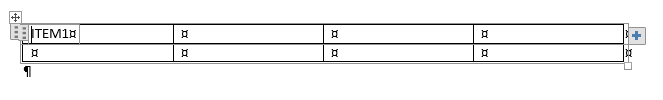
(Also, you could look at Properties (with or without Design Mode))

If you go into properties and add a repeating section title, you might expect it to appear when you hover over the “+” button.

It doesn’t. The only place it shows up is in your right click menu. (That’s as per Microsoft’s documentation)

## Tip #4: be aware that you might still wrestle with table row mapping

If you then you map lineitem[1]/productcode, you see:

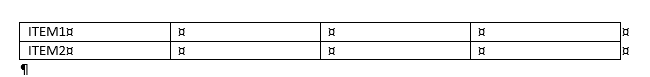


Note that if you go into that content control, it doesn’t highlight the corresponding element in the mapping pane. So there’s no visual feedback there.

More worringly, it hasn’t updated that second row, so you wonder whether things are working right.

Close & re-open the docx. Nope. Still the same. [strange .. I could’ve sworn that worked one time]

When things do work, its not until you close & re-open the docx that Word updates the number of rows to reflect your data:



If you do get it right, Word will insert the number of rows matching the number of items.

And it gets worse. If after you close and re-open the document, it still isn’t populated correctly to fix it, you need to manually delete those errant additional rows first. But this also deletes your data, so you can’t actually see your repeat working properly, until you use the “+” button, or re-populate your Custom XML part!

## Tip #5: map inside first

Microsoft suggested to me that things might work better if I map the item data first, then the repeat.

In other words:

* make a table row
* map lineitem[1]/productcode inside a table cell
* finally, select the table row and insert a repeating content control around it

It turns out this works much better. Done this way around, you get immediate feedback as your rows are populated appropriately.

# RepeatingSection positives

The first good thing about them is that they exist! That is, that Microsoft has added them in Word 2013.

With repeats, Microsoft is getting closer to being a complete standards based approach to document automation, which could replace the hundreds of incompatible solutions out there.

What’s missing:

* conditionals
* interactive Q/A

The second good thing about them is that the end user interface for adding/removing items is quite user friendly.

# RepeatingSection end-user negatives

We’ve explored/documented the problems with the developer interface above.

Here are some problems for end-users to be aware of (that is, people editing a docx containing bound production data).

The first problem is that “repeat0” doesn’t seem to work. Consider:

<inv:lines/>

Ie no line items.

Word 2013, when opening the docx, will still display a line item from last time (ie the document surface is not updated to correctly reflect the bound data!)

[would it be any different with an attached schema?]

The second problem is that Word 2010 breaks the docx, by stripping out the w15 namespace content control properties. That is, if a Word 2010 user saves the docx, the repeats won’t work any more. What’s worse, there is no warning to this effect (or of any type) when they open, or when they save.

Unfortunately (per [MS-DOCX]) this seems to be expected behaviour, since:

*“Word 2010 ignores a compatSetting element with a val attribute whose value is 15.”*

This leads into a third problem, which is the update lag in the installed base (2013 v 2010 v 2007). See for example, <http://www.networkworld.com/article/2225614/microsoft-subnet/office-2003-is-a-bigger-threat-to-microsoft-than-google-docs.html>

# Docx4j Roadmap

For users wanting to use w15 repeating section, docx4j (v3.2.2) will support updating the document surface from the bound Custom XML part.

However, until all these problems:

1. repeat0 deficiencies in baked in content control model
2. too many users on Word 2010 or 2007
3. Word 2010 silently breaks the document

are fixed, OpenDoPE repeats remain our recommended approach

In this document, all Word 2013 testing is with 15.0.4420.1017 (64-bit)