Convention for content control extensions for repeats and conditionals

Word's content control data binding provides a natural way to insert text, for example

Dear Click here to enter text.

That leaves the question of how to handle things like:

* **conditional inclusion** of paragraphs or other units of content
* **repeat** (eg of list items, table rows, or other units of content)
* inclusion of other documents as well (**altChunk**)

# Conditional Inclusion

A content control is said to be conditional if it (and its contents) are included/excluded from the document based on whether some condition is true or false.

Without a way to say that a content control is conditional, an XML file can't control whether paragraphs, tables etc appear in a document.

# Repeats

A content control is a repeat if it designates that its contents are to be included more than once.

For example, a row of a table for each invoice/order item, or person.

Without a way to say that a content control is to be repeated, an XML file can't contain variable amounts of repetitive content.

# Problem Statement

The problem is that the Open XML specification does not standardise how conditionals and repears can be done, and nor does Microsoft give any guidance or convention.

This is a significant limitation on document generation, which each system typically has to address.

A standardised way of doing repeats and conditionals would prevent businesses from re-inventing the wheel, and provide for enhanced interoperability.

The purpose of this document is to suggest a convention, which various tools could implement.

# Suggested Convention

The suggested convention is to include bindingrole=conditional|repeat in the Content Control's **tag**.

Putting the control information in the content control properties is a better design than putting it in the bound XML.

# Processing model

The binding role tag is preprocessed via an appropriate tool, to produce a new docx document.

Any content controls whose bindingrole=conditional evaluated to false will be missing from this new docx document.

Any content controls which had bindingrole=repeat will have their content appear n times, where n is the number of child nodes

# Example documents

An example can be found in <http://dev.plutext.org/svn/docx4j/trunk/docx4j/sample-docs/databinding/>

invoice.docx contains examples of conditionals and repeats, using the proposed conventions.

The custom xml used in the example is:

<invoice>

<customer>

<name>Joe Bloggs</name>

</customer>

<items>

<item>

<name>apples</name>

<price>$20</price>

</item>

<item>

<name>bananas</name>

<price>$30</price>

</item>

<item>

<name>cherries</name>

<price>$40</price>

</item>

<total>$90</total>

</items>

<misc>

<includeBankDetails>true</includeBankDetails>

</misc>

</invoice>

invoice\_preprocessed\_OUT.xml is the result of processing invoice.docx, using the docx4j implementation of this convention.

invoice\_bound\_OUT.xml is the result of processing all the binding information (ie the equivalent of what Word does when opening invoice\_preprocessed\_OUT.xml).

Notice that Word 2007 can open all 3 documents, and behaves as one would expect.

# bindingrole=conditional

The content control is excluded only if its databinding points to an XML element or attribute which has case-insensitive value "false"

If you look at invoice.docx, you'll see it contains an sdt with:

<w:sdtPr>

<w:tag w:val="bindingrole=conditional&amp;w:xpath=/invoice[1]/misc/includeBankDetails&amp;w:storeItemID={8b049945-9dfe-4726-9de9-cf5691e53858}" />

</w:sdtPr>

Notice that the information which would ordinarily be included in a w:dataBinding tag is instead encoded in the tag.

This approach ensures Word 2007 behaves as expected.

# Word Content Control Toolkit

The Content Control Toolkit is often used to set up data bindings. This program is easy to modify, so that if an element has a w:tag with w:val containing the word "bindingrole" then any binding information is encoded as above and not as a w:dataBinding element.

# bindingrole="repeat"

invoice.docx contains the following example:

<w:sdt>

<w:sdtPr>

<w:tag w:val="bindingrole=repeat  
 &amp;w:xpath=/invoice[1]/items  
 &amp;w:storeItemID={8b049945-9dfe-4726-9de9-cf5691e53858}" />

</w:sdtPr>

<w:sdtContent>

<w:tr>

<w:sdt>

<w:sdtPr>

<w:dataBinding w:xpath="/invoice[1]/items/item[1]/name"

w:storeItemID="{8B049945-9DFE-4726-9DE9-CF5691E53858}" />

<w:text />

</w:sdtPr>

<w:sdtContent>

<w:tc>

<w:p>

<w:r>

<w:t>apples</w:t>

</w:r>

</w:p>

</w:tc>

</w:sdtContent>

</w:sdt>

<w:sdt>

<w:sdtPr>

<w:dataBinding w:xpath="/invoice[1]/items/item[1]/price"

w:storeItemID="{8B049945-9DFE-4726-9DE9-CF5691E53858}" />

<w:text />

</w:sdtPr>

<w:sdtContent>

<w:tc>

<w:p>

<w:r>

<w:t>$20</w:t>

</w:r>

</w:p>

</w:tc>

</w:sdtContent>

</w:sdt>

</w:tr>

</w:sdtContent>

Here, the table row will be duplicated, once for each /invoice[1]/items/item

When the repeat is being processed, any w:dataBinding on any child sdt will need to be altered to point at the nth item.

A more sophisticated model would be to say it is cloned once for each child node that has certain specified name. For example, given:

<items>

<item>

<name>apples</name>

<price>$20</price>

</item>

<item>

<name>bananas</name>

<price>$30</price>

</item>

<item>

<name>cherries</name>

<price>$40</price>

</item>

<total>$90</total>

</items>

bindingrole=repeat[item] could produce a row for each item, and ignore the <total> node. Feedback is sought as to whether this flexibility is required.

# Implementation

This proposed convention is implemented in docx4j v2.5.0.

Source code can be found at <http://dev.plutext.org/svn/docx4j/trunk/docx4j/src/main/java/org/docx4j/openpackaging/parts/CustomXmlDataStoragePart.java>

# Namespace?

It is up to us to choose a namespace ([databindingconventions.org?](http://databindingconventions.org?))

Jason Harrop  
Plutext

document version 0.1  
15 July 2010