WordML2FO stylesheets documentation

Table of contents

Overview

The WordML2FO stylesheets documentation provides instructions for product installation and for converting WordProcessingML to XSL-FO using WordML2FO stylesheets. It includes information about implemented features, customization rules and limitations. It demonstrates how to render your own PDF file using WordML2FO stylesheets. The documentation also contains introduction to WordProcessingML and WordML2FO stylesheets, and glossary of basic terms.

WordProcessingML overview

Microsoft's WordprocessingML [http://www.microsoft.com/office/xml/default.mspx] is an XML representation, or schema, of the Word document file format. Microsoft Word 2003 can save documents in WordprocessingML by representing edited constructs in XML syntax. WordprocessingML is also known as WordML and Microsoft Office Word 2003 XML Reference Schemas. WordML is offered by Microsoft with open documentation and licenses.

WordML2FO overview

WordML2FO is a collection of stylesheets to convert WordProcessingML to XSL-FO [http://www.w3.org/Style/XSL/]. These stylesheets were prepared by RenderX's development team and Microsoft for general use. The team put together a generic stylesheet designed for performing this conversion, as well as an extension stylesheet that makes use of RenderX extensions such as flow-sections, document properties and bookmarks implemented in RenderX XEP formatter [http://www.renderx.com/tools/xep.html].
RenderX encourages any developer's interested in providing suggestions or enhancements to subscribe to our xep-support list [http://www.renderx.com/support/index.html] and submit your comments.

Installation

Software requirements

To prepare WordML documents and to create XSL-FO document Microsoft Word 2003 has to be installed on your computer.
To render PDF files from WordProcessingML using WordML2FO stylesheets you need to have XSL-FO formatter installed. If you would like to use RenderX specific extensions (such as flow-sections, document properties and bookmarks), you need RenderX XEP formatter [http://www.renderx.com/tools/xep.html] installed on your computer.

How to install WordML2FO

To install WordML2FO just extract archive of WordML2FO package to your computer with preserving subfolders structure.
After extracting the package WordML2FO is ready to use.

Inside product package

WordML2FO consists of stylesheets collection, documentation package, example set, readme file and license agreement.

Stylesheets

The collection of stylesheets located in "stylesheets" folder contains main stylesheets – generic and extension stylesheets to perform WordML to XSL-FO conversion – and auxiliary stylesheets with templates to translate WordML elements into their equivalents in XSL-FO.
You have to use only one of the main stylesheets (generic or extension) to perform WordML to XSL-FO conversion.
See detailed description of main and auxiliary stylesheets below.

Main stylesheets

- **Word2FO.xsl** converts WordProcessingML (Microsoft Word 2003 XML documents) to XSL-FO (XSL Formatting Objects). It can be used for any renderer, no extension elements are used.
- **Word2FO_ext.xsl** also converts WordProcessingML (Microsoft Word 2003 XML documents) to XSL-FO (XSL Formatting Objects). But it supports RenderX specific extensions such as flow-sections, document properties and bookmarks. It can be used efficiently with RenderX XEP formatter.

Auxiliary stylesheets

- **auxiliary.xsl** contains named templates to perform auxiliary tasks such as conversion of hexadecimal color values from Word to XSL-FO, substitution of Unicode by character codes in symbol fonts, normalization of font family names, parsing of 'CSS style'-like structures, page number format setting.
- **elementProperties.xsl** controls translation of WordML properties into their equivalents in XSL-FO.
- **elementStructure.xsl** contains templates to convert WordML structures into their equivalents in XSL-FO.
- **pageLayout.xsl** defines page layouts and generates XSL-FO page sequence based on WordML.
- **profile.xsl** adjusts the text layout of the XSL-FO document.
- **SmartTags.xsl** is a wrapper for implementing user stylesheet for processing smart tags of Word document.
- **WordArt.xsl** provides templates to convert WordArt elements into their equivalents in XSL-FO.
- Also stylesheets contain files to generate barcodes (these files are located in "Plugins\Barcodes" subfolder). To learn more about barcodes generation see documentation **RenderingBarcodes.pdf**.

Documentation

The documentation is located in “doc” folder and includes following files:

- **WordML2FOUserDocumentation.pdf** guides users around WordML2FO package; it is the file you are reading now.
- **RenderingBarcodes.pdf** describes how to generate barcodes using WordML2FO stylesheets.
- **WhatsNew.txt** lists improvements, fixed bugs, features and limitations for each release.
- **AddingPluginsToWordML2FOProject.pdf** explains how to include new smart-tag processing into original stylesheets set.

Examples

The set of WordML examples located in “examples” folder. You can get started with WordML2FO using these examples.

Readme

Readme file **Readme.txt** contains short description of WordML2FO package.

License

WordML2FO package is free and can be downloaded from RenderX website [http://www.renderx.com/download/index.html]. But it is important that you read license agreement document **License.txt** before using WordML2FO package.

How does WordML2FO work?

The generic stylesheet (**Word2FO.xsl**) is intended to be used with XSL-FO formatters. The stylesheet **Word2FO_ext.xsl** could be applied to add some features provided by RenderX extensions.

Element mapping

Following table describes the mapping of the WordML elements to their XSL-FO equivalents in WordML2FO stylesheets.

Document part	WordML tag	XSL-FO tag
Paragraph	w:p	fo:block
Text chunk (run element)	w:r	fo:inline
Bulleted and numbered lists	w:p (para with w:pPr/w:listPr)	fo:list-block, fo:list-item, fo:list-item-label, fo:list-item-body
Table	w:tbl, w:tr, w:tc	fo:table, fo:table-row, fo:table-cell
Image	w:pict	fo:external-graphic
Footnote	w:footnote/w:p	fo:footnote/fo:footnote-body
Hyperlink	w:hlink	fo:basic-link
Editor remarks	aml:annotation	fo:inline or fo:block
Tabulation	w:r/w:tab	Emulates through fo:leader (see limitations)

Implemented features

The generic WordML2FO stylesheet supports basic formatting objects such as text, paragraphs, numbered and bulleted lists, tables, text direction in table cells, headers and footers, styles, hyperlinks, section breaks, etc. Also the generic WordML2FO stylesheet offers rendering of following additional features:

- Footnotes support.
- Support of WordArt shapes and their properties (some features are not implemented yet, see limitations below for details. See also the file whatsnew.txt to get the information about new features availability).
- Ability to generate barcodes using smart-tags which can be inserted in a Word document.

In addition to listed above, extended WordML2FO stylesheet provides:

- Rendering of bookmarks tree based on heading outline level.
- Filling up document properties.
- Support of continuous sections with different number of columns.

WordML2FO stylesheets support rendering of WordML documents created with Microsoft Word 2003 to XSL-FO. Also there is a partial support of Microsoft Word 2007 documents.

Customization

WordML2FO collection of stylesheets offers possibility for third-party developers to add their specific processing of WordML file. They have to create their own XML schema and add template for processing specific tags. You can find more detailed information in documents **AddingPluginsToWordML2FO.pdf**, **RenderingBarcodes.pdf**.

Also you can control a bookmark tree rendering using Word2FO_ext.xsl by setting the parameters described below.

To manage a bookmark tree rendering in PDF file use parameter “default-show-bookmarks”. By default, “default-show-bookmarks” has value ‘true’ that is bookmark tree will be generated. To disable bookmark tree rendering set parameter “default-show-bookmarks” to ‘false’.

To manage auto numbering for items in generated bookmark tree in PDF file use parameter “default-numbered-pdf-bookmarks”. By default, “default-numbered-pdf-bookmarks” has value ‘true’ that is bookmark tree will be generated with auto numbering. To disable auto numbering for bookmark tree items in PDF file set parameter “default-numbered-pdf-bookmarks” to ‘false’.

There are several ways to set values of parameters described above:

1. Set parameters in profile.xsl stylesheet:
   <xsl:param name="default-show-bookmarks" select="'true"/>
   <xsl:param name="default-numbered-pdf-bookmarks" select="'true'"/>
2. Add parameters as WordML file properties:
   a. select “File” → “Properties” in menu WordML file

    b. switch to tab “Custom”
    c. type name of the parameter (“default-show-bookmarks” or “default-numbered-pdf-bookmarks”, without quotes), select type “Yes or no”, check “Yes” or “No”
    d. click “Add” button
    e. click “OK” button to apply changes to your WordML file (later you can change this parameter)

   Note: This way will override parameter values specified in the item 1.

3. Set XSLT stylesheet parameters for the transformation:
    a. default-show-bookmarks=true
    b. default-numbered-pdf-bookmarks=true

   Note: This way will override parameter values specified in the item 1 and 2.

Limitations

Formatting (on February 15th 2006)

- If you use localized version of the Microsoft Word to prepare WordML file, some features may be unsupported. It is caused by specific data storage usage by localized versions of the Microsoft Word.
- Limited list of fields is supported now: {PAGE}, {NUMPAGES} and {REF}. Some of specific fields can not be reproduced outside the Word at all. E.g. {Filesize}, {GoToButton}, etc.
- Sometimes text formatted with tabs looks distorted. XSL-FO is generated at the stage of preparation to render a document, so there is no adequate way to calculate width of text chunks. Thus there is no possibility (using only XSLT) to determine which tab-stop will be ignored or recovered from formatting mistake as Microsoft Word does.
- WordML file saved by Word 2007 as "Word 2003 XML document" contains no information for tabulation length. So, the default value 35.25pt is used instead. That's why resulting PDF document will look quite different than Word one formatted with tabulation stops. You may use tables with no borders as workaround to apply formatting similar to tabulation indentation.
- We do not support the hyphenation of text divided between several text parts. (Word produces text run for its own convenience, so it divides some words with complicated spelling into few runs.)
- We do not support frames.
- Pattern shading (using style and color) is not currently supported. You can use fill method of shading instead.
- The element cfChunk is not supported.
- If you use nested tables set formatting parameter “Repeat as header row at the top of each page” to OFF (unchecked) on table properties.
- WordML document should have top/bottom margin size greater or equal than document header/footer size (to provide more accurate top/bottom margin of XSL-FO).
- If bookmark is specified on the table row in WordML file, this bookmark will be placed at the beginning of the table.
- If text font size in table cell is specified by hand (not by style) and less than font size of current style applied, then table rows height will be rendered according to current style applied. To avoid this use style to set font settings for table cells.
- We do not support text effects like "Blinking Background".
- We do not render different page sizes and margins if WordML file consists of several continuous sections sequence with different page sizes and margins for pages from sequence. The page sizes and margins will be rendered according to the last one.
- Rendering of list’s bullets by Wingdings Font is supported only with limited number of characters and can be different visually a bit due to emulating Wingdings bullets by SVG.
- If list’s bullet is specified by several characters and Word does not save the actual indent to list’s item, the bullet may overlap this item. Suggested workaround is to specify the actual indent to list’s item manually in WordML document in “Bullets and Numbering” → “Customize” → “Tab space after”.
- We do not support processing of “Bullets and Numbering” → “Customize” → “Bullet position” – “Indent At:” and “Text position” – “Indent At:” values. Only “Text position” – “Tab space after:” processing is supported.
- If table width or table column width are specified in percent they may be rendered with a bit different width in XSL-FO. It is recommended to use exact measures.
- We do not support page layout with different width columns due to XSL-FO can support the columns of the same width only.

- Not all Wingdings and Symbol characters are supported.
- We do not support table alignment except left one due to XSL-FO specification.
- Text wrapping is not supported due to XSL-FO specification.
- In the case you set header height and document top margin to overlap text to produce special formatting (e.g. you want to make a decorative corner for document by inserting absolute positioned WordArt objects) header objects override document body text.
- We do not support table cell margins.
- We do not render document's background image and images inside Text Box.
- We do not support table styles made with "Table AutoFormat" Word feature.

Graphics (on February 15th 2006)

- If boundary of shape is out of drawing canvas, shape will be truncated by canvas rectangle in XSL-FO output.
- Lines with dotted or dash-dotted line style and width more than 3pt may be distorted. It is due to absence of an analog in SVG for WordArt dotted line style.
- The double (triple) line style is not supported by WordML2FO.
- Support of "Fill Effects" is not implemented for filling Auto-Shapes. Only solid filling with color is supported now.
- In some complicated cases WordArt objects may look a little bit differently in XSL-FO output.
- Text inside of WordArt objects:
  - Text orientation is not supported.
  - In case of WordArt shapes are on canvas, Z-ordering of text inside such shapes is not supported. Text is always rendered over the WordArt objects and can not lay under any of them.
  - For some complex WordArt objects inner text is rendered across the object instead of inside.
  - Text layout inside WordArt shapes may look differently: wrapping, cutting of bottommost line, etc.
- External BMP and TIFF images are not supported (external JPG/GIF and also embedded ones work fine).
- Arrows may look differently due to simulated rendering.
- WordArt shapes that are placed without drawing canvas will not be positioned correctly, unless their position is specified relative to margin.

Known Bugs (on February 15th 2006)

- The last continuous section is formatted using properties of previous section (with break for new page).
- Text paragraph indentation and spacing before/after in XSL-FO output differ a bit from original WordML.
- Incorrect processing of background image in WordML document with header that contained image.

How to create XSL-FO file using Microsoft Word

You can obtain an XSL-FO document from WordML document, using a third-party parser. Also, you can get XSL-FO directly from Microsoft Word:

- Open WordML document in Word.
- In the “Save As” dialog, check the Apply transform box.
  Note: ensure that the “Save data only” checkbox is cleared. If “Save data only” is checked when you apply the transform, Word discards the formatting stored in the document.
- Click “Transform”.
- In the “Choose an XML Transform” dialog, browse to the Word2FO.xsl stylesheet and click on “Open”.
  Note: the result will be a file with an XML extension.

How to render PDF file using WordML2FO stylesheets

You can render PDF file from WordML document using WordML2FO stylesheets and any XSL-FO formatter (result depends on how complete is the formatter's XSL-FO support).
To use RenderX extensions, you need RenderX XEP formatter [http://www.renderx.com/tools/xep.html].
See instructions below on rendering PDF file using WordML2FO stylesheets with XEP formatter. For other formatters your steps may be slightly different.
To obtain PDF file you need to perform the following actions:

Prepare WordML document

1. Open or create new document in Word.
2. Click menu item "File" → "Save As".
3. Choose "XML Document (*.xml)" in "Save as type" combo box (checkboxes "Apply transform" and "Save data only" must be cleared), enter file name and click Save.

Note: If you are using Microsoft Word 2007, choose "Word 2003 XML Document (*.xml)" in "Save as type" combo box. Please do not save file as "Word XML Document (*.xml)" (Word 2007 XML) - this format is not supported by WordML2FO stylesheets because of essential differences from WordML 2003 format.

Render PDF file from XML file and WordML2FO stylesheets

You can perform this either by "XEP Assistant" or from command line by executing XEP batch file. For more information about command line parameters, refer to the XEP documentation [http://www.renderx.com/reference.html].
To render PDF file using "XEP Assistant":

1. Run "XEP Assistant" ("<XEP installation directory>\x4u.bat", for example: "C:\Program Files\RenderX\XEP\x4u.bat").
2. Click menu item "File" → "Open", then choose your XML file and click "Open".
3. Click menu item "Formatting" → "Start".
4. In "Formatting settings" dialog check "Apply stylesheet", click "Browse" button and choose generic WordML2FO stylesheet file ("<WordML2FO installation directory>\stylesheets\Word2FO.xsl") or extended stylesheet file ("<WordML2FO installation directory>\stylesheets\Word2FO_ext.xsl").
   Note: <WordML2FO installation directory> means full path to the directory, where WordML2FO package is installed.
5. Click OK button.
6. If your PDF file is formatted successfully you can find it in directory that specified in "Formatting settings" dialog. Usually it is the same directory as in the path to XML file. If PDF file is not created – see "Event Log" tab in "XEP Assistant" with diagnostic. For more information, refer to XEP documentation [http://www.renderx.com/reference.html].

To preview resulting PDF you need a PDF viewer, for example Adobe Acrobat Reader [http://www.adobe.com/products/acrobat/readstep2.html].

References & Links

- Overview of WordprocessingML (Word 2003 XML Reference) [http://msdn.microsoft.com/library/en-us/WordXMLCDK/html/cdkPrimerPlaceholder_HV01113631.asp].
- Transforming Word Documents into the XSL-FO Format [http://msdn.microsoft.com/office/default.aspx?pull=/library/en-us/odc_wd2003_ta/html/OfficeWordWordMLtoXSL-FO.asp].
- Smart-tags in Microsoft Word [http://msdn.microsoft.com/library/default.asp?url=/msdnmag/issues/03/11/xmlfiles/toc.asp].
- Stylesheets and examples for rendering barcodes [http://renderx.com/demos/barcodes.html].
- Brian Jones: Open XML Formats [http://blogs.msdn.com/brian_jones/archive/2006/02/02/523469.aspx].

Glossary

SVG Scalable Vector Graphics (SVG) [http://www.w3.org/Graphics/SVG/] is an XML markup language for describing two-dimensional vector graphics, both static and animated. It is an open standard created by the World Wide Web Consortium. SVG 1.1 [http://www.w3.org/TR/SVG11/] is a W3C Recommendation and forms the core of the current SVG developments.

XEP RenderX XEP [http://www.renderx.com/tools/xep.html] is an XML to PDF (XSL FO) formatter. It takes input in XML, applies an XSL transformation to build XSL Formatting Objects representation, and then formats the Formatting Objects into PDF or PostScript. XEP supports multiple raster and vector graphic formats. Among them, Scalable Vector Graphics (SVG) is an XML-based vector graphics representation, widely used in business applications and for fine typesetting.

XSL-FO XSL-FO (XSL Formatting Objects) [http://www.w3.org/Style/XSL/] is an XML vocabulary for the formatting of documents. Being part of XSL, the normal way is to produce XSL-FO documents by transforming XML documents using XSLT. Even though the principles behind XSL and CSS (the other Style Sheet Language created by W3C) are quite different, it is planned to align the formatting model between XSL-FO and CSS, so that formatting engines can be based on the same code, both languages can be used to achieve the same results, and formatted results will look identical.

XSL XSL (Extensible Style Language) [http://www.w3.org/Style/XSL/] is a Style Sheet Language that can be used for displaying XML documents. Using XSL is two-step process, the first step being a transformation of the XML document using XSLT, and the second step being the rendering of the result of the transformation, which is done using XSL-FO. While XSL covers the same application area than CSS, it is much more powerful, because the transformation step (using XSLT) can perform arbitrarily complex transformations of the XML document, while CSS is not able to make any structural changes to the XML document.

XSLT XSLT (XSL Transformations) [http://www.w3.org/Style/XSL/] is a specialized Programming Language for transforming XML documents. Even though it is part of XSL and as such intended to be used for transforming XML documents into XSL-FO for presentation purposes, it is not limited to this application area. XSLT uses XML syntax (i.e., it is a Programming Language in XML syntax), even though it is based on DSSSL (which uses a Lisp-like syntax). XSLT is particularly interesting in B2B scenarios, where XML documents must be transformed.