

xtpxlib-xoffice

Conversions for Word and Excel files

0 Table of Contents

0 Xatapult XML Library - Conversions for Word and Excel files	2
1 Description	3
1.1 Converting from Excel (.xlsx)	3
1.2 Converting to Excel (.xlsx)	3
1.3 Converting from Word (.docx)	4
2 XProc Libraries	6
2.1 XProc (1.0) library: excel.mod.xpl	6
2.1.1 Step: xtlxo:extract-xlsx	6
2.1.2 Step: xtlxo:modify-xlsx	6
2.2 XProc (1.0) library: word.mod.xpl	6
2.2.1 Step: xtlxo:create-docx	6
2.2.2 Step: xtlxo:extract-docx	7
3 XML Schemas	8
3.1 XML Schema: xlsx-extract.xsd	8
3.2 XML Schema: xlsx-modify.xsd	8
4 XSLT Modules	9
4.1 XSLT (3.0): excel-conversions.mod.xsl	9
4.1.1 Function: xtlxo:excel-date-to-xs-date() as xs:date	9
4.2 XSLT (2.0): xoffice.mod.xsl	9
4.2.1 Named template: xtlxo:get-properties	11
4.2.2 Function: xtlxo:doc-href() as xs:string	11
4.2.3 Function: xtlxo:get-file-root() as element()?	11
4.2.4 Function: xtlxo:get-file-root-from-relationship-id() as element()?	11
4.2.5 Function: xtlxo:get-file-root-from-relationship-type() as element()?	11
4.2.6 Function: xtlxo:get-file-root-relationship() as element(mso-rels:Relationships)?	11
4.2.7 Function: xtlxo:get-href() as xs:string	11
4.2.8 Function: xtlxo:get-rels-href() as xs:string	11

0 Xatapult XML Library - Conversions for Word and Excel files



xtpplib library - component **xtpplib-xoffice** - **v1.1.A** (2020-02-16)
Xatapult Content Engineering - <http://www.xatapult.com> - +31 6 53260792
Erik Siegel - erik@xatapult.com

xtpplib-xoffice is part of the **xtpplib** library. **xtpplib** contains software for processing XML, using languages like XSLT and XProc. It consists of several separate components, all named **xtpplib-***. Everything can be found on GitHub (<https://github.com/xatapult>).

This component contains pipelines for converting Microsoft Office Word (.docx) and Excel (.xlsx) files to and from some more manageable XML formats.

Installation and usage information can be found on **xtpplib**'s main website <https://www.xtpplib.org>.

Technical information:

Component documentation: <https://xoffice.xtpplib.org>

License: GNU GENERAL PUBLIC LICENSE - Version 3, 29 June 2007

Git URI: [git@github.com:xatapult/xtpplib-xoffice](https://github.com/xatapult/xtpplib-xoffice).git

Git site: <https://github.com/xatapult/xtpplib-xoffice>

This component depends on:

- [xtpplib-container](#) (Support for XML containers (multiple files wrapped into one))
- [xtpplib-common](#) (Common component: Shared libraries and IDE support)

1 Description

Microsoft Office files are actually zip files with a lot of XML and other stuff inside. It is remarkably difficult to get to the actual contents of them: What is in Excel cell A1B2 or what is written in this Word document. To help with this, the xtpxlib-xoffice component contains XProc (1.0) pipelines to extract contents from Excel (.xlsx) and Word (.docx) files:

The namespace prefix `xtlxo:` is bound to the namespace `http://www.xtpxlib.nl/ns/xoffice` (`xmlns:xtlxo="http://www.xtpxlib.nl/ns/xoffice"`).

1.1 Converting from Excel (.xlsx)

The `xtlxo:extract-xlsx` pipeline takes an Excel .xlsx file and turns this into much more manageable XML. The schema for the resulting XML format is [here](#).

Take for instance this simple Excel sheet:

	1	2
1	1	What's up?
2	2	Cell with bold in it

Figure 1-1 - Excel example sheet

Running this through the `xtlxo:extract-xlsx` pipeline returns something like:

```
<?xml version="1.0" encoding="UTF-8"?>
<workbook xmlns="http://www.xtpxlib.nl/ns/xoffice"
  href="file:///path/to/excel.xlsx"
  timestamp="2019-12-11T12:50:20.252+01:00">

  <properties>
    ... Sheet properties ...
  </properties>

  <worksheet name="Sheet1">
    <row index="1">
      <cell index="1" ref="A1">
        <value>1</value>
      </cell>
      <cell index="2" ref="B1">
        <value>What's up?</value>
      </cell>
    </row>
    <row index="2">
      <cell index="1" ref="A2">
        <value>2</value>
        <formula>A1+1</formula>
      </cell>
      <cell index="2" ref="B2">
        <value>Cell with <span class="b">bold</span> in it</value>
      </cell>
    </row>
  </worksheet>
</workbook>
```

1.2 Converting to Excel (.xlsx)

The `xtlxo:modify-xlsx` pipeline takes a template Excel .xlsx file and changes this. The result will be written to a new Excel file.

It has the following features:

- You can change the individual worksheets in the Excel file. A worksheet is identified by its *name* (the name that is visible on its tab at the bottom of the Excel screen).
- You can identify a cell on a worksheet in three ways:
 - As a direct numeric row/column index
 - As identified by an Excel *name*. You can use this to identify a cell, by row, column, or both. An Excel name can reference an area (or even multiple areas) on a worksheet. To work around this the most upper-left cell in the named area(s) is used.
 - Using an Excel name (like above) and adding a numeric offset.

- You can insert a numeric or string value in a cell.
- You have to specify the type of the data to insert (so you can, for instance, insert a numeric value as a string if necessary)

There are some things you need to take care of creating the template Excel file:

- If you need formatting in a cell you're going to fill with this pipeline (like colors, borders, etc.) there *must* be some contents in the cell. Since this will be overwritten it should not be a problem.
- The same is true for a cell you're referencing by name: It must contain some contents. If you need this contents to be invisible you can always use a single space character.
- Names of worksheets and cells are case-sensitive.

The XML for specifying the changes to the Excel file is quite simple. The schema can be found [here](#). A simple example:

```
<xlsx-modifications xmlns="http://www.xtpxlib.nl/ns/xoffice">

  <worksheet name="TEST">

    <row name="NAMEDCELL" >
      <column name="NAMEDCELL" >
        <number>12345</number>
      </column>
      <column name="NAMEDCELL" offset="1">
        <string>One to the right</string>
      </column>
    </row>

    <row index="1">
      <column index="1">
        <string>Upper left-hand corner</string>
      </column>
      <column index="2">
        <number>6E3</number>
      </column>
    </row>
  </worksheet>

</xlsx-modifications>
```

1.3 Converting from Word (.docx)

The `xtpxlib:extract-docx` pipeline takes a Word (.docx) file and turns this into an understandable XML format. This format is more experimental than the format created by the Excel conversion and there isn't (yet) a schema for it.

As an example take this simple Word file:

Hello there!

Something in **Bold**!

- A list entry
- Another one

Simple table header	More header
Column1, row 2	Column 2 row 2

Figure 1-2 - Example Word document

Running this through the `xtpxlib:extract-docx` pipeline returns something like:

```

<document xmlns="http://www.xtpxlib.nl/ns/xoffice"
  dref=""
  timestamp="2019-12-11T13:09:15.415+01:00">
  <properties>
    ... document properties ...
  </properties>

  <p xml:space="preserve">Hello there!</p>
  <p xml:space="preserve">Something in <span class="b">Bold</span>!</p>
  <p class="ListBullet" xml:space="preserve">A list entry</p>
  <p class="ListBullet" xml:space="preserve">Another one</p>
  <p class="ListBullet" indent-left="360" indent-level="0" xml:space="preserve"/>
  <table>
    <tr>
      <td>
        <p class="ListBullet" indent-level="0" xml:space="preserve">Simple table header</p>
      </td>
      <td>
        <p class="ListBullet" indent-level="0" xml:space="preserve">More header</p>
      </td>
    </tr>
    <tr>
      <td>
        <p class="ListBullet" indent-level="0" xml:space="preserve">Column1, row 2</p>
      </td>
      <td>
        <p class="ListBullet" indent-level="0" xml:space="preserve">Column 2 row 2</p>
      </td>
    </tr>
  </table>
  <p class="ListBullet" indent-left="360" indent-level="0" xml:space="preserve"/>
</document>

```

There is an experimental pipeline [xtlxo:create-docx](#) to create Word documents (using a template Word document for things like styles, margins, etc.). If you feed this the same kind of XML you get from [xtpxlib:extract-docx](#), the result *should* be a valid, useable Word document with the new text in it. It's currently incomplete (it doesn't do tables for instance). Use at your own risk.

2 XProc Libraries

The xtpxlib-xoffice component contains the following XProc (1.0) library modules:

Module	Description
excel.mod.xpl	Conversions for Excel (.xlsx) files.
word.mod.xpl	Conversions for Word (.docx) documents.

Table 2-1 - Module overview

2.1 XProc (1.0) library: excel.mod.xpl

File: xplmod/excel.mod/excel.mod.xpl

Conversions for Excel (.xlsx) files.

Prefix	Namespace URI
xtlxo	http://www.xtpxlib.nl/ns/xoffice

Step	Description
xtlxo:extract-xlsx	Extracts the contents of an Excel (.xlsx) file in a more useable XML format .
xtlxo:modify-xlsx	Takes an input/template Excel (.xlsx) and a modification specification and from this creates a new modified Excel file that merges these two sources.

2.1.1 Step: xtlxo:extract-xlsx

Extracts the contents of an Excel (.xlsx) file in a more useable [XML format](#).

Port	Type	Primary?	Description
result	out	yes	The resulting XML representation of the Excel file.

Option	Rq?	Default	Description
xlsx-href	yes		Document reference of the .xlsx file to process (must have <code>file://</code> in front).

2.1.2 Step: xtlxo:modify-xlsx

Takes an input/template Excel (.xlsx) and a [modification specification](#) and from this creates a new modified Excel file that merges these two sources.

Port	Type	Primary?	Description
source	in	yes	The modification specification .
result	out	yes	The output is identical to the input but with <code>@timestamp</code> , <code>@xlsx-href-in</code> and <code>@xlsx-href-out</code> added to the root element.

Option	Rq?	Default	Description
xlsx-href-in	yes		URI of the input (template) .xlsx file to process
xlsx-href-out	yes		URI of the output .xlsx file.

2.2 XProc (1.0) library: word.mod.xpl

File: xplmod/word.mod/word.mod.xpl

Conversions for Word (.docx) documents.

Prefix	Namespace URI
xtlxo	http://www.xtpxlib.nl/ns/xoffice

Step	Description
xtlxo:create-docx	Turns Word XML (back) into a Word .docx file, using a template file.
xtlxo:extract-docx	Extracts the contents of a Word file in a more useable XML format.

2.2.1 Step: xtlxo:create-docx

Turns Word XML (back) into a Word .docx file, using a template file.

The input must be in the format the `xtlxo:extract-docx` pipeline creates.

Port	Type	Primary?	Description
source	in	yes	The Word XML that must be converted to <code>.docx</code> format.
result	out	yes	The document-container (see xtpxlib-container) as written to the final Word file.

Option	Rq?	Default	Description
result-docx-href	yes		Document reference where to write the resulting <code>.docx</code> file (must have <code>file://</code> in front).
template-docx-href	yes		Document reference of the template <code>.docx</code> file to use (must have <code>file://</code> in front).

2.2.2 Step: `xtlxo:extract-docx`

Extracts the contents of a Word file in a more useable XML format.

Port	Type	Primary?	Description
result	out	yes	The resulting XML representation of the Word file.

Option	Rq?	Default	Description
docx-href	yes		Document reference of the <code>.docx</code> file to process (must have <code>file://</code> in front).

3 XML Schemas

The xtpxlib-xoffice component contains the following XML Schemas:

Module	Description
xlsx-extract.xsd	Schema for the result of an Excel (.xlsx) data extraction to XML. Format produced by the xtlxo:extract-xlsx pipeline.
xlsx-modify.xsd	Schema for the modification specification of Excel (.xlsx) files. Format used by the xtlxo:modify-xlsx pipeline.

Table 3-1 - Module overview

3.1 XML Schema: xlsx-extract.xsd

File: xsd/xlsx-extract.xsd

Target namespace: <http://www.xtpxlib.nl/ns/xoffice>

Schema for the result of an Excel (.xlsx) data extraction to XML. Format produced by the [xtlxo:extract-xlsx](#) pipeline.

Element	Description
workbook	Root element of the Excel workbook extraction XML result.

3.2 XML Schema: xlsx-modify.xsd

File: xsd/xlsx-modify.xsd

Target namespace: <http://www.xtpxlib.nl/ns/xoffice>

Schema for the modification specification of Excel (.xlsx) files. Format used by the [xtlxo:modify-xlsx](#) pipeline.

Element	Description
xlsx-modifications	Root element of the Excel modifications specification.

4 XSLT Modules

The xtpxlib-xoffice component contains the following XSLT modules.

Module	Description
excel-conversions.mod.xsl	Excel data specific conversions
xoffice.mod.xsl	Library with support code for the MS Office file handling.

Table 4-1 - Module overview

4.1 XSLT (3.0): excel-conversions.mod.xsl

File: xslmod/excel-conversions.mod.xsl

Excel data specific conversions

Prefix	Namespace URI
xtlxo	http://www.xtpxlib.nl/ns/xoffice

Variable	Type	Value	Description
xtlxo:excel-start-date	xs:date	xs:date('1900-01-01')	

Function	Description
xtlxo:excel-date-to-xs-date()	Converts an Excel derived number into an xs:date

4.1.1 Function: xtlxo:excel-date-to-xs-date() as xs:date

Converts an Excel derived number into an xs:date

Parameter	Type	Description
excel-value	xs:anyAtomicType	The data coming from Excel. Will be cast to an integer. If this cannot be done the function returns the Excel start date, 1900-01-01.

4.2 XSLT (2.0): xoffice.mod.xsl

File: xslmod/xoffice.mod.xsl

Library with support code for the MS Office file handling.

Depends on the following XSLT modules from the xtpxlib-common component:

- general.mod.xsl
- href.mod.xsl

Yet largely undocumented. Use at your own risk.

Prefix	Namespace URI
xtlxo	http://www.xtpxlib.nl/ns/xoffice

Variable	Type	Value	Description
xtlxo:relationship-type-comments	xs:string	'http://schemas.openxmlformats.org/officeDocument/2006/relationships/comments'	
xtlxo:relationship-type-core-properties	xs:string	'http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties'	
xtlxo:relationship-type-custom-properties	xs:string	'http://schemas.openxmlformats.org/officeDocument/2006/relationships/custom-properties'	
xtlxo:relationship-type-extended-properties	xs:string	'http://schemas.openxmlformats.org/officeDocument/2006/relationships/extended-properties'	
xtlxo:relationship-type-main-document	xs:string	'http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument'	
xtlxo:relationship-type-shared-strings	xs:string	'http://schemas.openxmlformats.org/officeDocument/2006/relationships/sharedStrings'	

Named template	Description
xtlxo:get-properties	

Function	Description
xtlxo:doc-href()	
xtlxo:get-file-root()	
xtlxo:get-file-root-from-relationship-id()	
xtlxo:get-file-root-from-relationship-type()	
xtlxo:get-file-root-relationship()	
xtlxo:get-href()	
xtlxo:get-rels-href()	

4.2.1 Named template: xtlxo:get-properties

Parameter	Type	Rq?	Default	Description
extracted-office-xml	element(xtlcon:document-container)			

4.2.2 Function: xtlxo:doc-href() as xs:string

Parameter	Type	Description
href-parts	xs:string+	

4.2.3 Function: xtlxo:get-file-root() as element()?

Parameter	Type	Description
extracted-office-xml	element(xtlcon:document-container)	
href-parts	xs:string+	
is-mandatory	xs:boolean	

4.2.4 Function: xtlxo:get-file-root-from-relationship-id() as element()?

Parameter	Type	Description
extracted-office-xml	element(xtlcon:document-container)	
basefile-href	xs:string	
relationship-id	xs:string	
is-mandatory	xs:boolean	

4.2.5 Function: xtlxo:get-file-root-from-relationship-type() as element()?

Parameter	Type	Description
extracted-office-xml	element(xtlcon:document-container)	
basefile-href	xs:string	
relationship-type	xs:string	
is-mandatory	xs:boolean	

4.2.6 Function: xtlxo:get-file-root-relationship() as element(mso-rels:Relationships)?

Parameter	Type	Description
extracted-office-xml	element(xtlcon:document-container)	
basefile-href	xs:string	
is-mandatory	xs:boolean	

4.2.7 Function: xtlxo:get-href() as xs:string

Parameter	Type	Description
elm	element()	

4.2.8 Function: xtlxo:get-rels-href() as xs:string

Parameter	Type	Description
basefile-href	xs:string	