xtpxlib-xdoc

An xtpxlib component for generating documentation

Erik Siegel - Xatapult
2019-11-25

0 Table of Contents

0 Documentation generation with xtpxlib-xdoc 2
0.1 Installing xtpxlib-xdoc 2
0.2 Running xtpxlib-xdoc 3
1 Description 4
1.1 Overview 4
1.2 The xtpxlib-xdoc main toolchain 4
1.3 Documentation website generation 5
2 xdoc extensions 6
2.1 xdoc extension elements 6
2.2 Built-in xdoc transformations 6
2.2.1 XProc 1.0 pipeline: code-docgen.xpl 6
2.2.2 XProc 1.0 pipeline: xml-description.xpl 6

0 Documentation generation with xtpxlib-xdoc

Parameter	Value
DATE	2019-11-25
DATETIME	2019-11-25 11:54:28
HREF-SOURCE	C:/Data/Erik/work/xatapult/xtpxlib-xdoc/doc/source/xtpxlib-xdoc-componentdoc.xml
TBD-DELETE-LATER	bla bla bla
TIME	11:54:28
author-email-address	erik@xatapult.com
author-name	Erik Siegel
component-dependencies	xtpxlib-common xtpxlib-container
component-name	xtpxlib-xdoc
core-component-name	xtpxlib-common
current-release-date	2019-11-11
current-release-version	1.0
git-uri	git@github.com:eriksiegel/xtpxlib-xdoc.git
github-pages-uri	http://www.xatapult.nl
library-name	xtpxlib
owner-company-github	https://github.com/xatapult
owner-company-name	Xatapult Content Engineering
owner-company-phone	+31 6 53260792
owner-company-website	http://www.xatapult.com

Table 0-1 - Parameters

Component **`xtpxlib-xdoc`** version 1.0 - 2019-11-11

Xatapult Content Engineering - http://www.xatapult.com - +31 6 53260792

Erik Siegel - erik@xatapult.com

`xtpxlib-xdoc` is a component for generating documentation. It uses DocBook 5.1 + additional markup as its base format and can (currently) generate PDF and XHTML as output. It can also generate component documentation (like the one you're looking at now), as a simple website and PDF. See "Description" on page 4 for an overview.

`xtpxlib-xdoc` is part of the **`xtpxlib`** library. **`xtpxlib`** contains software for processing XML, using languages like XSLT and XProc. It consists of several separate components, named `xtpxlib-*`. Everything can be found on GitHub (https://github.com/xatapult). The core component of `xtpxlib` is `xtpxlib-common`. Most components will be accompanied by a GitHub pages documentation site (TBD).

0.1 Installing xtpxlib-xdoc

To install **`xtpxlib-xdoc`** you can do one of the following:

- Clone its GitHub repository (git@github.com:eriksiegel/xtpxlib-xdoc.git) to some appropriate location on disk. The `master` branch will always contain the latest stable version (currently 1.0 2019-11-11).
- Or download the latest release from https://github.com/xatapult/xtpxlib-xdoc/releases and unpack it somewhere.

`xtpxlib-xdoc` is dependent on the following other `xtpxlib` components. **Important:** These must be installed in the *same base directory* as **`xtpxlib-xdoc`**:

- `xtpxlib-common`
- `xtpxlib-container`

0.2 Running xtpxlib-xdoc

`xtpxlib-xdoc` consists of XProc (1.0) pipelines and XSLT (2.0 and 3.0) stylesheets. To run these you'll need:

- To run the XProc pipelines use XML Calabash Xproc 1.0 processor. This library uses several of its non-standard extensions, another XProc processor is therefore probably not be usable.
- To run XSLT stylesheets the Saxon processor is preferred. In most cases the (open source) HE (Home Edition) version will be sufficient. If not a PE (Professional Edition) license is necessary.
- As an alternative run the software from within the oXygen XML IDE.

1 Description

1.1 Overview

`xtpxlib-xdoc`'s main points:

- **`xtpxlib-xdoc`**'s starting point is documentation written in DocBook 5.1 plus extensions. This source format is called *xdoc*.
- `xdoc` has two kinds of extensions:
  - Parameters, coming from a parameter file, that are expanded. This useful for, for instance, status information, dates/times, standard words and phrases, etc. [TBD LINK]
  - The so-called **[*** Referenced linkend id "chapter-xdoc-transforms" not found (phase: inline)]**. These transformations convert something into DocBook and insert the result back in the main document. This is used for, for instance code documentation generation. There are standard transformations but you can also write your own (in XSLT or XProc).
- An `xdoc` document can be converted into pure DocBook by an XProc pipeline [TBD LINK]
- Additional XProc pipelines [TBD LINK] convert this DocBook into PDF (through XSL-FO) or basic XHTML.
- This mechanism also serves as a basis for documentation generation of complete software components [TBD LINK]. The result's are meant to be used as GitHub pages.
- Noteworthy additional functionality is an XProc module [TBD LINK] that converts (simple) Markdown into DocBook.

1.2 The xtpxlib-xdoc main toolchain

The following figure illustrates **`xtpxlib-xdoc`**'s main toolchain:

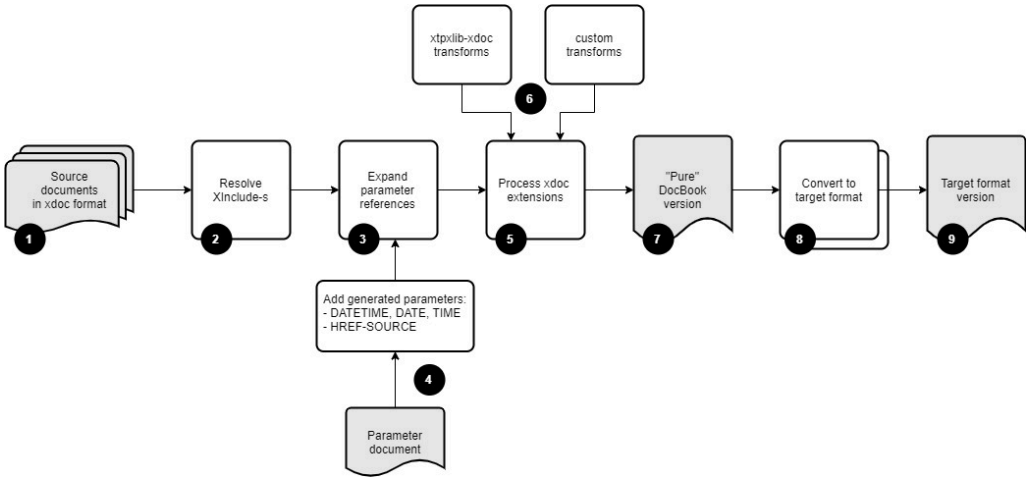

*Figure 1-1 - **`xtpxlib-xdoc`**'s main toolchain*

1. The **`xtpxlib-xdoc`** module uses a format called **`xdoc`** as its source format. The basis of `xdoc` is a limited version of DocBook 5.1. On top of this `xdoc` adds several extensions for parameter handling, generating text, etc.

   The DocBook 5.1 standard is huge and therefore for practical reasons only partly implemented in this component. A description of what is supported can be found in [TBD].
2. The first processing step in the toolchain performs basic XInclude processing. This means that you can build your document from smaller parts, for instance one file per chapter.

   Another application of the XInclude processing is to get the data in for the `xdoc` extensions processing in step 5.
3. The next step is to expand any parameter references in the source document. A parameter is a name/value pair. To expand its value in the document use either `${name}` or `{$name}` (both mean the same). Parameters are expanded both in text and in attribute values.

4. Parameters come from two sources:
   - An (optional) parameter file. This file must be in the parameters format as handled by the parameters module of `xtpxlib-common`. For instance:

     ```
     <parameters>
       <parameter name="my-parameter">
         <value>Some value…</value>
       </parameter>
     </parameters>
     ```

     This format has several additional features. See [TBD] for more information.
   - The toolchain adds several parameter of its own:

Parameter	Description	Example value(s)
`DATETIME`	The date and time the toolchain executed in `YYYY-MM-DD hh:mm:ss` format.	`2019-11-11 12:12:12`
`DATE`	The date part of the `DATETIME` parameter.	`2019-11-11`
`TIME`	The time part of the `DATETIME` parameter.	`12:12:12`
`HREF-SOURCE`	The main source's filename.	`C:/my/path/sourcedoc.xml` `/my/path/sourcedoc.xml`

*Table 1-1 - Parameters added by the **`xtpxlib-xdoc`** toolchain*

5. Next the so-called `xdoc` extensions are processed. These extensions are specific elements in the `xdoc` (`http://www.xtpxlib.nl/ns/xdoc`) namespace. Currently there are two:

   `<xdoc:transform>`
   The `<xdoc:transform>` element triggers a transformation with this element (and its children) as input. Transformations can be either XProc (1.0) or XSLT (2.0 or 3.0). The result5 of such a transformation must be valid DocBook and is inserted in the document.

   `<xdoc:dump-parameters>`
   This extensions dumps all parameters in the document, either as a table or as an XML comment. Useful for debugging the parameter expansion. See [TBD]

6. The transformations triggered by `<xdoc:transform>` can come from two sources:
   - Transformations that are built into the **`xtpxlib-xdoc`** module. These are generic transformations for, for instance, documenting XML structures or code. An overview of these can be found in [TBD].
   - Your own transformations. Guidelines on how to write these can found in [TBD].
7. The result of the toolchain so-far is a document in "pure" DocBook 5.1.
8. From this you can transform to some target format. The **`xtpxlib-xdoc`** component contains transformations to PDF and HTML (see [TBD]). These transformations are rather specific for the **`xtpxlib-xdoc`** component and might not be directly usable for other use-cases. You can of course copy-and-adapt these or use some other DocBook conversion transformation .
9. The result of all this is a document in the desired target format.

Information about the pipelines that implement this toolchain can be found in [TBD]

1.3 Documentation website generation

[TBD]

2 xdoc extensions

The **xtpxlib-xdoc** component uses (a subset of) DocBook 5.1 as its main vocabulary. Within this there can be so-called *xdoc extensions*).

- xdoc extensions are specific elements in the xdoc (http://www.xtpxlib.nl/ns/xdoc) namespace. These are described in "xdoc extension elements" on page 6.

2.1 xdoc extension elements

[TBD]

2.2 Built-in xdoc transformations

2.2.1 XProc 1.0 pipeline: code-docgen.xpl

This pipeline takes some document (XSL, XSD, ordinary XML, etc.), wrapped inside an xdoc:transform element, and searches it for documentation. It then tries to auto-generate documentation in the intermediate docgen format (see xsd/docgen-intermediate.xsd).

When the format has the means to add documentation of itself (like for XProc and XML Schema), this is used. When there is no such thing (like for XSLT and straight XML), comments starting with a tilde (~) are used.

The descriptions can contain simple Markdown.

You can use an @filecomponents attribute on the xdoc:transform element to control the way the filename is displayed:

- When lt 0, no filename is displayed
- When eq 0, the full filename (with full path) is displayed
- When gt 0, this number of filename components is displayed. So 1 means filename only, 2 means filename and direct foldername, etc.

TBD: @header-level

Port	Tp.	Pr?	Description
source	in	*	The document to generate documentation for, wrapped in an xdoc:transform element. TBD also works without wrapper
result	out	*	The resulting docbook 5 output

2.2.2 XProc 1.0 pipeline: xml-description.xpl

Turns an XML element description into the appropriate DocBook. A schema for the element's description markup can be found in xsd/element-description.xml.

Typical usage (within an xdoc source document):

```
<xdoc:transform href="$xdoc/xml-description/xml-description.xpl>
  <xi:include href="path/to/xml/description.xml"/>
</xdoc:transform>
```

Port	Tp.	Pr?	Description
source	in	*	The document containing the XML description, wrapped in an xdoc:transform element.
result	out	*	The resulting DocBook output, containing the element's description