

xtpplib

An XML processing library

0 Table of Contents

0 XML processing xtpplib library	2
1 Description	3
2 Usage	4
2.1 Installing xtpplib components	4
2.2 Running xtpplib components	4

0 XML processing xtpplib library



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

Erik Siegel - erik@xatapult.com

xtpplib is an open source library containing 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>). See its [description](#) for more information. Installation and usage instructions can be found [here](#).

The Xatapult XML Library currently consists of the following components:

Component	Description
xtpplib-common	Common component: Shared libraries and IDE support
xtpplib-container	Support for XML containers (multiple files wrapped into one)
xtpplib-xoffice	Conversions between Word/Excel and XML
xtpplib-xdoc	DocBook publication toolchain

1 Description

The `xtpxlib` library is an open source library containing software for processing XML, using languages like XSLT and XProc. It consists of several separate [components](#). I've tried to document everything on the accompanying GitHub pages.

Some highlights (the parts I use most often):

- The `xtpxlib-common` component contains several [XSLT modules](#). For instance:
 - `general.mod.xsl` contains a lot of small utility functions and templates. For instance capitalize a string, safe conversions from strings to other data types, etc.
 - `href.mod.xsl` is a library I use very often. It contains all kinds of href (filename or web address) juggling functions. For instance get the name/extension/path portion of an href, get/replace/delete its protocol, canonicalize an href (replace the `.` and `..` parts, compute relative href-s, etc.
 - `parameters.mod.xsl` contains code for the general handling of parameters and the substitution of parameter values in texts. There is a separate [parameters documentation section](#) about this.
- The <https://container.xtpxlib.org/> component contains code for handling *XML containers*: Constructs that provide support for working with multiple related files by wrapping them into a single one. Binary files are referenced instead of included. Also a very useful technique if you need to create a lot of files (like when you're generating a website).
- The <https://xoffice.xtpxlib.org/> component contains an [XProc pipeline](#) for turning Excel sheets into much more legible and processable XML. This allows Excel to be used as a user interface to data-driven XML applications.
- The `xtpxlib-xdoc` component contains a [toolchain](#) for DocBook driven documentation. Among its functionality are [pipelines](#) with mechanisms for generating DocBook from other markup. It also contains custom PDF and HTML generating.

2 Usage

2.1 Installing xtpplib components

To install an `xtpplib` component, first go to its documentation home page to find details about where it is on GitHub and the Git URI you need for cloning it:

Component	Description
xtpplib-common	Common component: Shared libraries and IDE support
xtpplib-container	Support for XML containers (multiple files wrapped into one)
xtpplib-xoffice	Conversions between Word/Excel and XML
xtpplib-xdoc	DocBook publication toolchain

Then do one of the following:

- Clone its GitHub repository to some appropriate location on disk. The `master` branch will always contain the latest stable version.
- Or download the latest release zip from its git website (releases page) and unpack it somewhere.

That's all folks...

2.2 Running xtpplib components

`xtpplib` consists mainly 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 the [XML Calabash](#) Xproc 1.0 processor. The `xtpplib` library uses several of its non-standard extensions, another XProc processor is therefore not (directly) 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](#).