xtpxlib-container

XML Container Handling

Erik Siegel - Xatapult Content Engineering
2019-12-10

0 Table of Contents

0 Xatapult XML Library - XML Container Handling

`xtpxlib` library - component **`xtpxlib-container`** - v0.0 2019-12-10
Xatapult Content Engineering - http://www.xatapult.com - +31 6 53260792
Erik Siegel - erik@xatapult.com

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

XML Containers provide support for working with multiple related files by wrapping them into a single one. Binary files are referenced instead of included.

The container structure is standardized. Once contents is in a container it's easy to analyze, change and/or write back. It can also be used to create a whole file structure, in a container, and then write it out to disk or zip file.

The `xtpxlib-container` component has XProc (1.0) pipelines for:

- Reading the contents of a zip file or directory structure into a container
- Writing a container out to a zip file or disk

Installation and usage information can be found on **`xtpxlib`**'s main website https://www.xtpxlib.org.

<u>Technical information:</u>

Component documentation: https://container.xtpxlib.org

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

Git URI: `git@github.com:xatapult/xtpxlib-container.git`

Git site: https://github.com/xatapult/xtpxlib-container

This component depends on:

- xtpxlib-common (Common component: Shared libraries and IDE support)

1 Description

An XML container as handled/used by this module is an XML structure that holds other XML documents and references to binary files. Here is a short example:

```
TBD
```

Why is this useful? Here are some use-cases:

- Office docs
- Creating a website dir structure

1.1 The container format

The root of an XML container document is an `<xtlcon:document-container>` element. The prefix `xtlcon:` must be bound to the namespace `http://www.xtpxlib.nl/ns/container` (`xmlns:xtlcon="http://www.xtpxlib.nl/ns/container"`).

```
<xtlcon:document-container timestamp = xs:dateTime
                           href-source-zip? = xs:string
                           href-target-zip? = xs:string
                           href-target-zip-result? = xs:string
                           href-target-zip-tmpdir? = xs:string
                           href-source-path? = xs:string
                           href-target-path? = xs:string
                           href-target-result-path? = xs:string
                           (any)? >
  ( <xtlcon:document> |
    <xtlcon:external-document> |
    <(any)> )*
</xtlcon:document-container>
```

Attribute	#	Type	Description
`timestamp`	1	`xs:dateTime`	The timestamp when this container was initially created/generated.
`href-source-zip`	?	`xs:string`	When the container was read from a zip file (using `xtlcon:zip-to-container`), this attribute holds the href of this zip file.
`href-target-zip`	?	`xs:string`	Holds the name of the zip file for writing the container to (using `xtlcon:container-to-zip`).
`href-target-zip-result`	?	`xs:string`	After the container is written to a zip file using **`[*** Referenced linkend id "2container.mod.xpl-xtlcon_container-to-zip" not found (phase: inline)]`**, this attribute will hold the full canonical filename of the zip file.
`href-target-zip-tmpdir`	?	`xs:string`	After the container is written to a zip file using `xtlcon:container-to-zip`, this attribute will hold the full canonical name of the temporary directory used for this process (probably not very useful).
`href-source-path`	?	`xs:string`	When the container was read from a directory (using `xtlcon:directory-to-container`), this attribute holds the href of this directory.
`href-target-path`	?	`xs:string`	Holds the name of the directory for writing the container to (using `xtlcon:container-to-disk`).
`href-target-result-path`	?	`xs:string`	After the container is written to a directory file using `xtlcon:container-to-disk`, this attribute will hold the full canonical name of the directory.
`(any)`	?		Any other attributes are allowed, so additional information can be added for use during processing.

Child element	#	Description
`xtlcon:document`	*	A document inside the container structure. See "Internal documents" on page 4.
`xtlcon:external-document`	*	An external document, referenced from the container structure. See "External documents" on page 4.
`(any)`	*	Any other elements are allowed, so additional information can be added for use during processing.

1.1.1 Internal documents

An *internal* document is a document whose contents is inside the container document. This will in most cases be XML documents, but text is also possible. It must be surrounded by an `<xtlcon:document>` element:

```
<xtlcon:document href-source? = xs:string
                 href-source-result? = xs:string
                 href-target? = xs:string
                 href-target-result? = xs:string
                 mime-type? = xs:string
                 (any)? >
  <(any)>
</xtlcon:document>
```

Attribute	#	Type	Description
`href-source`	?	`xs:string`	href of the source for this document. When this document comes from a zip file, it holds the href of the file *in* the zip.
`href-source-result`	?	`xs:string`	After processing holds the full canonical name of the source file.
`href-target`	?	`xs:string`	href of the target for this document.
`href-target-result`	?	`xs:string`	After processing holds the full canonical name of the target file.
`mime-type`	?	`xs:string`	Some specific values for this attribute trigger special conversions on output. See "Document MIME type values" on page 5.
`(any)`	?		Any other attributes are allowed, so additional information can be added for use during processing.

Child element	#	Description
`(any)`	1	Root element + contents of the document.

1.1.2 External documents

An *external* document is a document that is only referenced from the container. Usually binary files but anything goes. The referencing is done using an `<xtlcon:external-document>` element:

```
<xtlcon:external-document (attributes-from-internal-document)?
                          href-source-zip? = xs:string
                          href-source-zip-result? = xs:string
                          not-in-global-source-zip? = xs:boolean >
  <(any)>
</xtlcon:external-document>
```

All attributes of an internal document plus the following:

Attribute	#	Type	Description
`(attributes-from-internal-document)`	?		See internal documents.
`href-source-zip`	?	`xs:string`	Reference to the source zip file for this document. If present overrides `/*/@href-source-zip`
`href-source-zip-result`	?	`xs:string`	After processing holds the full canonical name of the source zip file.
`not-in-global-source-zip`	?	`xs:boolean`	Default: `false` When set to `true`, the global zip file `/*/@href-source-zip` is not used. This is necessary to allow references to external files that, when a global zip file is used, come from elsewhere.

Child element	#	Description
`(any)`	1	Any other elements are allowed, so additional information can be added for use during processing.

1.1.3 Document MIME type values

TBD

2 XProc Libraries

The xtpxlib-container component contains the following XProc (1.0) library module:

Module	Description
`container.mod.xpl`	XProc library with steps for handling xtpxlib containers.

Table 2-1 - Module overview

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

File: `xplmod/container.mod/container.mod.xpl`
XProc library with steps for handling xtpxlib containers.

Prefix	Namespace URI
`xtlcon`	`http://www.xtpxlib.nl/ns/container`

Step	Description
`xtlcon:container-to-disk`	Writes the contents of a container to disk.
`xtlcon:container-to-zip`	Writes the contents of a container to a zip file.
`xtlcon:directory-to-container`	Reads a directory into a container. All XML files will be read into the container, all other files will be included/referenced as external contents.
`xtlcon:zip-to-container`	Reads a zip file into a container. All XML files will be read into the container, all other files will be included/referenced as external contents.

2.1.1 Step: xtlcon:container-to-disk

Writes the contents of a container to disk.

Port	Type	Primary?	Description
`source`	in	yes	The container to process.
`result`	out	yes	The input container, but with changes that reflect the writing process.

Option	Rq?	Default	Description
`href-fop-config`		`resolve-uri('../../../xtpxlib-common/data/fop-default-config.xml', static-base-uri())`	Optional reference to an Apache FOP configuration file. Must be absolute! When not present a default file will be used.
`href-target`		`''`	Base path where to write the container. When you specify this it will have precedence over a /*/@href-target-path.
`indent-xml`		`false()`	Whether to indent the XML we create or not.
`remove-target`		`true()`	Whether to attempt to remove the target directory before writing.

2.1.2 Step: xtlcon:container-to-zip

Writes the contents of a container to a zip file.

Port	Type	Primary?	Description
`source`	in	yes	The container to process.
`result`	out	yes	The input container, but with all the changes in links, paths, etc.

Option	Rq?	Default	Description
`href-fop-config`		`''`	Optional reference to an Apache FOP configuration file. Must be absolute! When not present a default file will be used.
`href-target-zip`			Base path where to write the container. When you specify this it will have precedence over /*/@href-target-zip.
`indent-xml`		`false()`	Whether to indent the XML we create or not.

2.1.3 Step: xtlcon:directory-to-container

Reads a directory into a container. All XML files will be read into the container, all other files will be included/referenced as external contents.

Port	Type	Primary?	Description
`result`	out	yes	The output container.

Option	Rq?	Default	Description
`add-document-target-paths`		`true()`	Adds (relative) source paths as the target paths to the individual documents.
`href-source-directory`	yes		Reference to the directory to read.
`href-target-path`		`''`	Optional target path to record on the container.

2.1.4 Step: xtlcon:zip-to-container

Reads a zip file into a container. All XML files will be read into the container, all other files will be included/referenced as external contents.

Port	Type	Primary?	Description
`result`	out	yes	The output container.

Option	Rq?	Default	Description
`add-document-target-paths`		`true()`	Adds source paths as the target paths to the individual documents.
`href-source-zip`	yes		Reference to the zip file to read.
`href-target-path`		`''`	Optional target path to record on the container.

3 XML Schemas

The xtpxlib-container component contains the following XML Schemas:

Module	Description
`container.xsd`	Schema for an XML container.

Table 3-1 - Module overview

3.1 XML Schema: container.xsd

File: `xsd/container.xsd`
Target namespace: `http://www.xtpxlib.nl/ns/container`
Schema for an XML container.

Element	Description
`document-container`	Root element for a document container.