

---

# The xmltool command-line utility

Hussein Shafie

Published July 21, 2008

## Abstract

This document is the reference manual of the **xmltool** command-line utility. This utility, like all the other command-line utilities, is found in `XXE_install_dir/bin/`.

|  |   |
|--|---|
| 1. Synopsis .....                        | 1 |
| 2. <code>validate</code> options .....   | 1 |
| 3. <code>indent</code> options .....     | 2 |
| 4. <code>schematron</code> options ..... | 5 |
| 5. <code>schemadoc</code> options .....  | 5 |
| 6. Common options .....                  | 6 |

## 1. Synopsis

Usage:

```
xmltool validate|indent|schematron|schemadoc ?options? arguments
```

The **xmltool** utility comprises 4 different processors:

`validate` [1]

Checks the validity of a document conforming to a DTD, W3C XML Schema or RELAX NG schema.

May also be used to check the validity of a DTD, W3C XML Schema or RELAX NG schema.

`indent` [2]

Saves one or more documents after reformatting their XML contents.

May be also used to *flatten* these documents, that is, to transclude XInclude elements or DITA conrefs.

`schematron` [5]

Checks the validity of a document against a Schematron schema.

May also be used to check the validity of a Schematron schema.

`schemadoc` [5]

Generates a reference manual in HTML format for a DTD, W3C XML Schema or RELAX NG schema.

The generated HTML reference manual, organized like "*DocBook: The Definitive Guide*" by Norman Walsh and al., lists all the elements and attributes specified in the schema.

This manual is intended to help content authors create instances conforming to a given schema. This manual is not intended to help schema authors document their design.

Note that, for now, this documentation generator cannot extract documentation contained in a schema (i.e. in annotation/documentation elements) and merge extracted documentation with automatically generated documentation.

## 2. `validate` options

Usage:

```
xmltool validate validate_options common_options [6] [ xml_file ]*
```

Validates specified XML files. If no XML files are specified, it is the schemas specified using the "-s" option which are validated.

-s *schema*

Use specified schema file to validate all specified XML files.

*schema* must be a DTD having a ".dtd" extension or a W3C XML Schema having a ".xsd", ".xs", ".wxs" extension or a RELAX NG schema having a ".rng", ".rnc" (compact syntax) extension.

By default, the schema used for validation is found in each specified XML file (e.g. <!DOCTYPE>).

It is possible to specify several "-s" options:

- When no XML files are specified, each schema is individually validated.
- When several XML files are specified, the schemas are composed to form a compound schema (e.g. DocBook 5+MathML) and this compound schema is used to validate specified XML files.

-f

When validating a document, ignore false duplicate ID errors caused by multiple inclusions of the same element. When saving a document, replace such false duplicate IDs by automatically generated ones.

Example: Validate `docbook-image.xml` against the DTD specified in its <!DOCTYPE>.

```
/opt/xxe/demo$ xmltool validate docbook-image.xml
```

Example: Validate `docbook-table.xml` and `docbook-image.xml` against the DocBook 4.5 DTD.

```
/opt/xxe/demo$ xmltool validate -s ../addon/config/docbook/dtd/V4.5/docbookx.dtd \  
docbook-table.xml docbook-image.xml
```

Example: Validate the DocBook 4.5 DTD.

```
/opt/xxe/demo$ xmltool validate -s ../addon/config/docbook/dtd/V4.5/docbookx.dtd
```

Example: Validate `sample.xml` against the combined `docbook.rng` and `mathml2.rng` RELAX NG schemas.

```
/opt/xxe/addon/mathml_config/db5mml$ xmltool validate \  
-s ../../config/docbook5/rng/V5.0/docbook.rng \  
-s rng/mathml2.rng \  
sample.xml
```

Note that the combined schemas don't need to be of the same kind:

```
/opt/xxe/addon/mathml_config/db5mml$ xmltool validate \  
-s ../../config/docbook5/rng/V5.0/docbook.rng \  
-s ../standalone/xsd/mathml2.xsd \  
sample.xml
```

### 3. `indent` options

Usage:

```
xmltool indent indent_options validate_options [1] common_options [6] [ xml_file ]*
```

Save, possibly indenting and/or flattening<sup>1</sup>, specified XML files.

-o *save\_file\_or\_dir*

Specifies where to save all specified documents.

If a single document is to be saved, *save\_file\_or\_dir* may specify a file or an existing directory.

---

<sup>1</sup>That is, transclude references.

If multiple documents are to be saved, *save\_file\_or\_dir* must specify a directory. Such directory is automatically created if it does not already exist.

Default: reuse the original filename of each specified document after renaming this document using a ".BAK" filename.

**-flat**

Do not preserve inclusions. Instead ``flatten" a document.

Default: preserve inclusions.

**-indent** *integer*

Specifies the number of space characters used to indent a child element relatively to its parent element.

- A positive or null value means: indent always.
- Value "-1" means: never indent.
- Any other negative value means indent, but only if the document to be saved has an actual document type. In such case, the number of space characters is:  $(-2 - integer)$ .

Default: -4.

**-maxlinelength** *positive\_integer*

Specifies the maximum line length for elements containing text interspersed with child elements.

Default: 78.

**-noopenlines**

Do not add open lines between the child elements of elements having an ``element-only" content model.

Default: add open lines.

**-nolegacy**

Do no attempt to create an XML file which favors the interoperability with HTML/SGML.

Default: favor interoperability with HTML/SGML.

**-nocharentities**

Do not save characters not supported by the encoding as entity references. Instead, save them as numeric references.

Default: when possible and when needed to, save characters as entity references.

**-specialchars** *list\_of\_chars\_or\_char\_ranges*

Always save specified characters as entity references.

Example: `-specialchars "reg 174 0x00ae 0256 pound:yen 163:165 0xA3:0xA5 0243:0245"`.

No default.

**-cdatasections** *list\_of\_simple\_XPaths*

Save the textual contents of specified elements as CDATA sections.

XHTML example: `-cdatasections "htm:script htm:style"`.

No default.

**-prefix** *prefix namespace*

Associates a prefix to a namespace.

Multiple "-prefix" options are allowed.

This may be needed to allow parsing the XPaths argument of the above "-cdatasections" option. Options "-prefix" must precede the "-cdatasections" option.

XHTML example: `-prefix htm http://www.w3.org/1999/xhtml`.

No default.

**-nooriginalprefixes**

Do not use the namespace prefixes originally specified in the document. Instead, generate prefixes.

Default: Reuse the original prefixes as much as possible.

**-nodefaultnamespace**

Do not use the default namespace originally specified in the document.

Default: Reuse the default namespace if any.

**-xmlversion 1.0|1.1|original**

Specifies the "version" pseudo-attribute of the XML declaration. "original" means: reuse the XML version originally specified in the document.

Default: `original`.

**-encoding java\_supported\_encoding|original**

Specifies which encoding to use when saving a document. "original" means: reuse the encoding originally specified in the "encoding" pseudo-attribute of the XML declaration of the document.

Default: `original`.

**-standalone yes|no|original**

Specifies the "standalone" pseudo-attribute of the XML declaration. "original" means: reuse the "standalone" pseudo-attribute originally specified in the document.

No default: do not add a "standalone" pseudo-attribute to the XML declaration.

**-noinvalid**

Do not save specified documents if any of them is found to have validity errors (even harmless cross-reference errors).

Default: save documents even if some of them are found to be invalid.

Example: Indent `docbook-table.xml` using the default settings. The original `docbook-table.xml` is saved to `docbook-table.xml.BAK`.

```
/opt/xxe/demo$ xmltool indent docbook-table.xml
/opt/xxe/demo$ ls docbook-table.xml*
docbook-table.xml
docbook-table.xml.BAK
```

Example: Indent `docbook-table.xml` using specified settings. Save indented file to `out.xml`.

```
/opt/xxe/demo$ xmltool indent -indent 1 -noopenlines -nolegacy -o out.xml docbook-table.xml
```

Example: Force the indentation of schema-less file `xhtml_strict.xxe`.

```
/opt/xxe/addon/config/xhtml$ xmltool indent -indent 2 -o indented.xxe xhtml_strict.xxe
WARNING: Cannot determine which schema to use for validating "xhtml_strict.xxe".
```

Example: Indent all `.xhtml` files contained in current directory. Create save files in directory `/tmp/out/`.

```
/opt/xxe/demo$ xmltool indent -v [6] -encoding Windows-1252 -o /tmp/out *.xhtml
```

Example: Transclude all XInclude elements contained `docbook-modular-book.xml` (`-f [2]` is needed otherwise `out.xml` would contain a number of duplicate ID errors).

```
/opt/xxe/demo$ xmltool indent -xi [6] -f [2] -flat -o out.xml docbook-modular-book.xml
```

## 4. **schematron** options

Usage:

```
xmltool schematron schematron_options common_options [6] schematron [ xml_file ]*
```

Validate specified XML documents against specified Schematron.

Optionally validate the Schematron itself.

Unless the `-iso` option is used, the Schematron may be embedded in another type of XML document (e.g. a DocBook 5 RELAX NG grammar not using the compact syntax).

`-iso`

Fully validate the Schematron as an ISO Schematron schema.

Default: do not validate the Schematron, just load it.

Note that the Schematron loader is very lenient and accepts ISO Schematron as well as Schematron 1.5 schemas.

`-o out_schematron_file`

Save the Schematron to specified file. The written schema is in all cases an ISO Schematron schema using the minimal syntax.

`-phase phase_id`

Specifies the ID of the phase which is to be used for validation. May also be `#ALL` or `#DEFAULT`.

Default: `#DEFAULT`, if any, `#ALL` otherwise.

`-var name value`

Specify overrides for some of the let variables defined in the Schematron.

Note that value must be a valid XPath expression and not a plain string.

Example: Validate `docbook.sch` as an ISO Schematron schema. Additionally save a copy in `/tmp/out.sch`.

```
/opt/xxe/addon/config/docbook$ xmltool schematron -iso -o /tmp/out.sch docbook.sch
```

Example: Validate `docbook-image.xml` against `docbook.sch`.

```
/opt/xxe/demo$ xmltool schematron ../addon/config/docbook/docbook.sch \
docbook-image.xml
```

Example: Validate `docbook-image.xml` against `docbook.sch`, using phase `#ALL`. Pass to the Schematron variable `foo` having XPath string literal `"bar"` as its value.

```
/opt/xxe/demo$ xmltool schematron -phase '#ALL' -var foo '"bar"' \
../addon/config/docbook/docbook.sch \
docbook-image.xml
```

## 5. **schemadoc** options

Usage:

```
xmltool schemadoc schemadoc_options common_options [6] schema doc_dir
```

Generate in directory `doc_dir` a reference manual in HTML format for schema file `schema`.

*Schema* must be a DTD having a ".dtd" extension or a W3C XML Schema having a ".xsd", ".xs" or ".wxs" extension or a RELAX NG schema having a ".rng" or ".rnc" (compact syntax) extension.

*doc\_dir* is automatically created if it does not exist.

-css *css\_url*

Specifies the URL of the CSS style sheet used for the generated HTML.

Default: generated HTML does not have a `<link rel="stylesheet">`.

-charset *encoding*

Specifies the charset used for the generated HTML.

Default: use platform default encoding and generated HTML does not have a `<meta http-equiv="Content-Type">`.

-xxe

Generate annotations which are meaningful when using the schema with XMLmind XML Editor.

Example:

```
/opt/xxe/demo$ xmltool schemadoc bugreport/bugreport.xsd /tmp/bugreport_doc
/opt/xxe/demo$ ls /tmp/bugreport_doc
a__6sv96.html
abbr__79rrv.html
...
index.html
...
workaround__jus1.html
```

## 6. Common options

-xi

When loading a document, transclude XInclude elements.

Default: XInclude elements are treated as any other element.

This option is equivalent to "`-inclscheme com.xmlmind.xml.xinclude.XIncludeScheme`".

-inclscheme *class\_name*

When loading a document, transclude nodes specifying an inclusion directive belonging to specified inclusion scheme.

Default: nodes specifying an inclusion directive are treated as any other node.

Specifying several `-inclscheme` options is permitted. Mixing `-xi` and `-inclscheme` options is permitted.

-cache *schema\_cache\_dir*

Specifies the directory to be used as a schema cache. This directory is automatically created if it does not exist.

Default: do not cache schemas.

RELAX NG schemas can be cached only in memory and not on disk.

-rncencoding *encoding*

Specifies the encoding used for RELAX NG compact syntax schemas.

Default: do not cache schemas.

-v

Verbose.

Example: Transclude all XInclude elements contained docbook-modular-book.xml (-f [2] is needed otherwise out.xml would contain a number of duplicate ID errors).

```
/opt/xxe/demo$ xmltool indent -xi -f [2] -flat -o out.xml docbook-modular-book.xml
```

Example: Transclude all conref elements contained in map\_reference.dita.

```
/opt/xxe/addon/dita_dtd_config/doc_src$ CLASSPATH="../dita.jar" \  
XML_CATALOG_FILES="../dita_catalog.xml" \  
xmltool indent -inclscheme "com.xmlmind.xmleditext.dita.ConrefScheme" \  
-f -flat -o out.dita map_reference.dita
```

Example: Validate docbook-table.xml. Cache the DocBook DTD if it is not already cached. If it is already cached, use the cached copy.

```
/opt/xxe/demo$ xmltool validate -cache /tmp/cache docbook-table.xml  
/opt/xxe/demo$ ls /tmp/cache  
directory.txt  
docbookx.ser
```

Example: Validate mathml.pane against pane.rnc, a RELAX NG schema using the compact syntax, encoded in ISO-8859-1.

```
/opt/xxe/addon/mathml_config/common/pane$ xmltool validate \  
-rncencoding ISO-8859-1 -s pane.rnc mathml.pane
```