
The **xxetool** command-line utility

Hussein Shafie, XMLmind Software <xmlmind-support@xmlmind.com>

May 22, 2018

Abstract

This document is the reference manual of the **xxetool** command-line utility. This tool is basically the XMLmind XML Editor desktop application, but without a graphical user interface (GUI). Its main use is to convert XML documents to other formats (HTML, PDF, RTF, etc) from within a script or a makefile.

This utility, like all the other command-line utilities, is found in `xxe_install_dir/bin/`.

Table of Contents

1. Why use the xxetool command-line utility?	1
2. Using xxetool	1
2.1. The online documentation of xxetool	1
2.2. Using xxetool to convert documents	2
2.3. Using xxetool to compare documents	6
A. About xxetool and user authentication	8

1. Why use the **xxetool** command-line utility?

This tool is essentially XMLmind XML Editor (XXE for short) desktop application, *but without a graphical user interface* (GUI). Its main use is to convert XML documents to other formats (HTML, PDF, RTF, etc) from within a script or a makefile. It can also be used to compare two revisions of the same document in which change detection has been activated in *XMLmind XML Editor - Online Help*.

Because **xxetool** is almost identical to XXE, this command-line utility will automatically make use of all the installed add-ons (e.g. "Apache FOP 1.x XSL-FO processor plug-in", "Apache Batik image toolkit plug-in", etc) and also of the last specified user preferences (e.g. credentials allowing to access remote storage facilities).

This utility, like all the other command-line utilities, is found in `xxe_install_dir/bin/`¹.

Note

The **xxetool** command-line tool will not run on a headless server.

On the other hand, you can use **xxetool** to convert documents stored on a remote server (HTTP, WebDAV, FTP, ssh, etc). More information in Appendix A, *About **xxetool** and user authentication* [8].

2. Using **xxetool**

2.1. The online documentation of **xxetool**

xxetool is fully auto-documented:

- Run "`xxetool`" to list all available "functions":

¹If you install XXE on the Mac using the recommended `.dmg` distribution, you'll not find `xxe_install_dir/bin/`. This utility, like all the other command-line utilities, is found in `XMLmind.app/Contents/Resources/xxe/bin/`.

```
$ xxetool
usage: xxetool [Advanced Option]* convert|compare Options
```

- Run "xxetool convert" to list all available process commands:

```
$ xxetool convert
db5.toEclipseHelp
db5.toEpub
db5.toHTML
...
xhtml.toPSPrinter
xhtml.toRTF
```

- Run "xxetool convert dita.toEPUB" to print how to use the "dita.toEPUB" process command:

```
$ xxetool convert dita.toEPUB
xxetool convert [options] dita.toEPUB map_or_topic_file_or_URL \
-u epub_file_or_URL

Convert DITA map or topic map_or_topic_file_or_URL to
EPUB file epub_file_or_URL.

Example:
xxetool convert dita.toEPUB doc.ditamap -u out/doc.epub
```

- Run "xxetool convert -?" to print how to use the "convert function":

```
$ xxetool convert -?
usage: xxetool [Advanced Option]* convert [Other Option]*
[-t XSLT_stylesheet_file_or_URL]?
[-r|-ru resource_name resource_value]*
[-p XSLT_stylesheet_param_name XSLT_stylesheet_param_value]*
process_command_name doc_file_or_URL
[-s|-u process_command_arg]*

Converts XML document doc_file_or_URL using process command
...
```

- Run "xxetool compare -?", or more simply "xxetool compare", to print how to use the "compare function":

```
$ xxetool compare
usage: xxetool [Advanced Option]* compare [Other Option]* Options
-----
Options
-----
-on [ file_or_URL ]+
  Enable the comparison of revisions.
  file_or_URL may specify a local directory, in which case
  this directory is recursively traversed.
...
```

2.2. Using xxetool to convert documents

Usage:

```
xxetool [Advanced Option]* convert [Other Option]*
[-t XSLT_stylesheet_file_or_URL]?
[-r|-ru resource_name resource_value]*
[-p|-pu XSLT_stylesheet_param_name XSLT_stylesheet_param_value]*
```

```
process_command_name doc_file_or_URL
[-s|-u process_command_arg]*
```

Converts XML document *doc_file_or_URL* using process command in *XMLmind XML Editor - Commands* called *process_command_name*, found in any of the XXE configuration files scanned during the startup of **xxetool** (see *XMLmind XML Editor - Configuration and Deployment*).

Options:

`-t XSLT_stylesheet_file_or_URL`

Use this alternate XSLT style sheet instead of the one specified in the first `transform` child element of the process command.

If specified process command has no `transform` child element but has `subProcess` child elements, these sub-processes are searched recursively for a `transform` child element.

`-r|-ru resource_name resource_value`

Copy specified resource rather than the one specified in the `<copyProcessResources name="resource_name">` child element of the process command.

`-ru` is useful when the resource value is a relative filename that needs to be converted to an absolute "file:" URL.

`-p|-pu XSLT_stylesheet_param_name XSLT_stylesheet_param_value`

Add/replace corresponding XSLT style sheet parameter in the first `transform` child element of the process command.

`-pu` is useful when the parameter value is a relative filename that needs to be converted to an absolute "file:" URL.

If specified process command has no `transform` child element but has `subProcess` child elements, these sub-processes are searched recursively for a `transform` child element.

`-p|-pu param_group_name/param_name param_value`

Add/replace specified parameter to parameter group called *param_group_name*.

`-pu` is useful when the parameter value is a relative filename that needs to be converted to an absolute "file:" URL.

DocBook example: `-p docb.toRTF.XFCParameters/docx.variant 152`, where parameter group `docb.toRTF.XFCParameters` is declared as follows in `XXE_install_dir/addon/config/docbook/xslMenu.incl`:

```
<command name="docb.toRTF">
  <process>
    ...
    <processFO processor="XFC" file="__doc.fo" to="__doc.%0">
      <parameter name="outputFormat">%0</parameter>
      <parameter name="outputEncoding">%1</parameter>
      <parameter name="imageResolution">120</parameter>
      <parameter name="prescaleImages">>false</parameter>
      <parameterGroup name="docb.toRTF.XFCParameters" />
    </processFO>
    ...
  </process>
</command>
```

²This XMLmind XSL-FO Converter option marks generated DOCX file as being compatible with MS-Word 2013.

`-s` | `-u` *process_command_arg*

Pass these arguments to the process command as the values of process variables %0, %1, ..., %9.

If `-s` (String) is specified, the argument is passed as is.

If `-u` (URL) is specified, the argument, a file or directory name, is first converted to an URL.

Other options:

`-profile` | *file_or_URL*

Apply this conditional processing (profiling) file to the document being converted.

This file may be

- a ".ditaval" file optionally ending with "?media=screen" (default media) or "?media=print"
- OR a ".profiles" file ending with a fragment (e.g. "#my_profile") specifying the ID of the selected profile.

Specify "-" to suppress profiling.

`-d`

Sets the `debug` attribute of the process command to value `true` (no matter what has been specified in the `process` element).

This prevents the process command from deleting its work directory (`/tmp/xxeNNNN/`) at the end of the processing.

`-v` | `-vv` | `-vvv`

Turn verbosity on. The more Vs, the more verbose.

Advanced options:

`-auth` *credentials*

This option can be used to specify authentication credentials for a given server. This allows to connect to the specified server without interactively asking the user to enter a username and a password.

String *credentials* consists in 6 fields: *host*, *port*, *prompt*, *scheme*, *username*, *password*, in that order, separated by a newline character ('\n'). Fields *host*, *port*, *prompt*, *scheme* can be left empty, which means: match any. The UTF-8 bytes of the string are then encoded in base-64.

Command-line utility `xxe_install_dir/bin/authvalue` allows to generate such encoded string. Example: `encode string "\n\nDocument Store\n\nvictoria\n\nsecret"`:

```
/opt/xxe/bin$ authvalue victoria secret - "Document Store"
CgpEb2N1bWVudCBTdG9yZQoKanZpY3RvcmlhCnNlY3JldA==

/opt/xxe/bin$ xxetool convert -auth CgpEb2N1bWVudCBTdG9yZQoKanZpY3RvcmlhCnNlY3JldA== \
docb.toHTML http://www.acme.com/docstore/push_up.xml -u docs/
```

Command-line utility **authvalue** is auto-documented. Type **authvalue**, then press Enter to display a short documentation explaining how to use this utility.

`-putpref` *key value*

Adds or replace preference specified by *key/value* to the set of the user's preferences.

Note that the `-putpref`, `-putprefs`, `-delprefs` options change the user's preferences only during this invocation of **xxetool**. **xxetool** reads, but never writes the contents of file `XXE_user_preferences_dir/preferences.properties`.

`-putprefs property_file_or_URL`

Similar to `-putpref` except that several `key/value` pairs may be read from specified property file.

`-delpref key`

Removes preference specified by `key` from the set of the user's preferences.

Examples:

Example 1. Convert a DocBook document to multi-page HTML

1. First run "xxetool convert" to list all available process commands:

```
$ xxetool convert
db5.toEclipseHelp
db5.toEpub
...
docb.toHTML
...
```

2. Process command `docb.toHTML`³ looks good. Run "xxetool convert docb.toHTML" to print its online help:

```
$ xxetool convert docb.toHTML

xxetool convert [options] docb.toHTML docbook_file_or_URL \
-u output_dir_filename_or_URL

Convert DocBook document docbook_file_or_URL to multi-page HTML.
Create the HTML pages in directory output_dir_filename_or_URL.

Example:
xxetool convert docb.toHTML doc.xml -u out/
```

3. Use process command `docb.toHTML` to convert DocBook document `help.xml` to multi-page HTML created in directory `docs/help/`.

```
$ xxetool convert -p toc.section.depth 4 -p chunk.section.depth 2 \
docb.toHTML help.xml \
-u docs/help
```

Example 2. Convert a DocBook document to PDF

1. First run "xxetool convert" to list all available process commands:

```
$ xxetool convert
db5.toEclipseHelp
db5.toEpub
...
docb.toPSFile
...
```

³The `docb.toHTML` process command is defined in `XXE_install_dir/addon/config/docbook/xslMenu.incl`.

2. Process command `docb.toPSfile`⁴ looks good. Run "`xxetool convert docb.toPSfile`" to print its online help:

```
$ xxetool convert docb.toPSfile

xxetool convert [options] docb.toPSfile docbook_file_or_URL \
  -s pdf -s "|pdf" -u pdf_file_or_URL

Convert DocBook document docbook_file_or_URL to PDF file pdf_file_or_URL.

Example:
xxetool convert docb.toPSfile doc.xml -s pdf -s "|pdf" -u out/doc.pdf
...
```

3. Use process command `docb.toPSfile` to convert DocBook document `doc.xml` to `commands.pdf`.

```
$ xxetool convert -t fo_docbook.xsl \
  -p toc.section.depth 4 -p callout.graphics 0 -p variablelist.as.blocks 1 \
  docb.toPSfile doc.xml \
  -s pdf -s "|pdf" -u docs/commands/commands.pdf
```

Notice that an alternate, customized, XSLT style sheet, `fo_docbook.xsl`, is used instead of the stock `docbook.xsl`.

2.3. Using **xxetool** to compare documents

Usage:

```
xxetool [Advanced Option]* compare [Other Option]* Options
  -on [file_or_URL]+ |
  -off [file_or_URL]+ |
  -revon [file_or_URL]+ |
  -revoff [file_or_URL]+ |
  -addrev out_file file_or_URL [file_or_URL]+ |
  -cmp|-diff file_or_URL1 file_or_URL2
```

Options:

`-on [file_or_URL]+`

Enable the comparison of revisions in all specified documents. This implies adding `<?xxe-sn>` processing-instructions to all elements and then saving the modified document to disk.

No effect if comparison of revisions is already enabled in a document.

file_or_URL may specify a *local* directory, in which case this directory is recursively traversed.

`-off [file_or_URL]+`

Do not keep any information about revisions in all specified documents. This implies removing all `<?xxe-sn>` processing-instructions and possibly also the `<?xxe-revisions>` processing-instruction from the document and then saving the modified document to disk.

No effect if a document already does not contain any information about revisions.

file_or_URL may specify a *local* directory, in which case this directory is recursively traversed.

Note that option `-off` implies option `-revoff`; no need to specify both.

⁴The `docb.toPSfile` process command is defined in `xxe_install_dir/addon/config/docbook/xslMenu.incl`.

`-revon [file_or_URL]+`

Start storing revisions in all specified XML documents. This implies adding `<?xxe-sn>` processing-instructions to all elements, adding a single `<?xxe-revisions>` processing-instruction at the end of a document and then saving the modified document to disk.

No effect if a document already stores the revision history.

`file_or_URL` may specify a *local* directory, in which case this directory is recursively traversed.

Note that option `-revon` implies option `-on`; no need to specify both.

`-revoff [file_or_URL]+`

Stop storing revisions in all specified XML documents. This implies removing the `<?xxe-revisions>` processing-instruction found at the end of the document and then saving the modified document to disk.

No effect if a document already does not store the revision history.

`file_or_URL` may specify a *local* directory, in which case this directory is recursively traversed.

`-addrrev out_file file_or_URL [file_or_URL]+`

All specified `file_or_URL` files being revisions of the same document, store in `out_file` all these revisions. Specified `file_or_URL` files are automatically sorted by their date. File `out_file` is basically a copy of the most recent revision.

`-cmp file_or_URL1 file_or_URL2`

Compare specified documents. If these documents differ, print the changes on the console.

The documents being compared must be two revisions of the same initial document in which the comparison of revisions has been enabled.

The arguments may specify two *local* directories, in which case these directories are recursively traversed in parallel.

`-diff file_or_URL1 file_or_URL2`

Same as `-cmp` except that changes are not detailed. If one or more documents differ, **xxetool** simply exits with a non-zero exit code.

Other options:

`-v|-vv|-vvv`

Turn verbosity on. The more Vs, the more verbose.

Advanced options:

`-auth credentials`

This option can be used to specify authentication credentials for a given server. This allows to connect to the specified server without interactively asking the user to enter a username and a password.

String `credentials` consists in 6 fields: `host`, `port`, `prompt`, `scheme`, `username`, `password`, in that order, separated by a newline character (`'\n'`). Fields `host`, `port`, `prompt`, `scheme` can be left empty, which means: match any. The UTF-8 bytes of the string are then encoded in base-64.

Command-line utility `xxe_install_dir/bin/authvalue` allows to generate such encoded string. Example: encode string `"\n\nDocument Store\n\nvictoria\n\nsecret"`:

```
/opt/xxe/bin$ authvalue victoria secret - "Document Store"
CgpEb2N1bWVudCBTdG9yZQoKanZpY3RvcmlhCnNlY3JldA==

/opt/xxe/bin$ xxetool convert -auth CgpEb2N1bWVudCBTdG9yZQoKanZpY3RvcmlhCnNlY3JldA== \
docb.toHTML http://www.acme.com/docstore/push_up.xml -u docs/
```

Command-line utility **authvalue** is auto-documented. Type **authvalue**, then press Enter to display a short documentation explaining how to use this utility.

`-putpref key value`

Adds or replace preference specified by *key/value* to the set of the user's preferences.

Note that the `-putpref`, `-putprefs`, `-delprefs` options change the user's preferences only during this invocation of **xxetool**. **xxetool** reads, but never writes the contents of file `XXE_user_preferences_dir/preferences.properties`.

`-putprefs property_file_or_URL`

Similar to `-putpref` except that several *key/value* pairs may be read from specified property file.

`-delpref key`

Removes preference specified by *key* from the set of the user's preferences.

A. About **xxetool** and user authentication

Most remote document repositories (e.g. WebDAV servers) will require the user of **xxetool** to authenticate herself/himself.

xxetool being a command-line tool designed to be used in makefiles, batch files, shell scripts, etc, authentication by the means of an interactive dialog with the user is not a solution. That's why **xxetool** automatically uses the credentials stored by XMLmind XML Editor in the preferences file of the user.

Example:

User john needs to run **xxetool** to convert `http://www.acme.com/docs/foo.xml` to HTML.

Server `http://www.acme.com/docs/` requires user john to authenticate himself.

User john starts XMLmind XML Editor and opens `http://www.acme.com/docs/foo.xml`.

A dialog box is displayed prompting user john for his credentials.

User john types his user name and password and, by checking the "Remember these user name and password" checkbox, allows XMLmind XML Editor to store the credentials in `XXE_user_preferences_dir/preferences.properties`.

From then, user john may run **xxetool** to convert any document stored on `http://www.acme.com/docs/`.

See also the `-auth` command-line option [4] for an alternative way to let the user of **xxetool** authenticate to a server.