
XMLmind XML Editor - XHTML Support

Hussein Shafie, Pixware <xmlmind-support@xmlmind.com>

September 29, 2011

1. Supported DTDs	1
1.1. Omitting the XML declaration from the save file	2
2. XHTML menu	2
2.1. Convert Document sub-menu	3
3. Custom bindings	4
4. Table rendering	5
5. Spell checking	5
A. Parameters of the XSLT stylesheets used to convert XHTML to XSL-FO	5

1. Supported DTDs

XMLmind XML Editor (XXE for short) supports the following XHTML DTDs:

XHTML 1.0 Strict

The *Strict DTD*, which is content-oriented.

<!DOCTYPE> declaration added by XXE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

XHTML 1.0 Transitional

The *Transitional DTD*, which is compatible with traditional HTML.

<!DOCTYPE> declaration added by XXE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

XHTML 1.1

This document type is similar to XHTML 1.0 Strict, just slightly simpler and cleaner. However, like XHTML 1.0 Transitional, it makes it possible specifying attribute `target` on the `a` element. Example: `XHTML 1.1`.

<!DOCTYPE> declaration added by XXE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Important

if you have created your XHTML document using an editor which does not add one of the three above <!DOCTYPE> declarations at the top of your document, XXE will fail to recognize your document as being an XHTML one.

When this is the case, you'll have to fix this problem by adding one of the three above <!DOCTYPE> declarations, by hand, using a text editor. If you don't know which <!DOCTYPE> declaration corresponds to your document, then choose XHTML 1.0 Transitional [1] which is the most lenient one.

1.1. Omitting the XML declaration from the save file

Omitting the XML declaration (that is, `<?xml version="1.x" ...?>`) is useful when an XHTML document is delivered by the Web server to the Web browser as if it were an HTML document. That is, for the Web browser, the media type of the document is `text/html` and not `application/xhtml+xml`.

This is useful because both the XML declaration and the `<!DOCTYPE>` declaration have an effect on the behavior of Web browsers. See *Activating Browser Modes with Doctype*.

- If the document does not contain `<meta http-equiv="Content-Type" ... />`, then the XML declaration is *not* omitted.
- If you want to omit the XML declaration from the save file, then add `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />` to the head element.

Note that:

- For the XML declaration to be omitted, the media type must be `"text/html"` and the charset must be `"UTF-8"`.
- Element `<meta charset="CHARSET" />` is considered to be equivalent to `<meta http-equiv="Content-Type" content="text/html; charset=CHARSET" />`.
- The values of all the aforementioned attributes are parsed in a case-insensitive manner.
- If you just want to force the encoding of a specific XHTML document to be, for example, Windows-1250 without having to tweak Options → Preferences, Save options, then add `<meta http-equiv="Content-Type" content="application/xhtml+xml; charset=Windows-1250" />` to the head element.

2. XHTML menu

Table editing commands fully support HTML 4 tables. Most table editing commands can be repeated by using Edit → Repeat (**Ctrl+A**).

Menu	Item	Description
Column For a command in this menu to work, click anywhere inside a cell (or explicitly select a cell or an element having a cell ancestor).	Insert Before	Insert a column before column containing specified cell.
	Insert After	Insert a column after column containing specified cell.
	Cut	Cut to the clipboard the column containing specified cell.
	Copy	Copy to the clipboard the column containing specified cell.
	Paste Before	Paste copied or cut column before column containing specified cell.
	Paste After	Paste copied or cut column after column containing specified cell.
	Delete	Delete the column containing specified cell.
Row For a command in this menu to work, click anywhere inside a cell (or explicitly select a cell or an element having a cell ancestor) or explicitly select a row.	Insert Before	Insert a row before row containing specified cell. Note Note that row editing commands are enabled, not only by implicitly or explicitly selecting a table cell or any of its descendants, but also by explicitly selecting a table row.
	Insert After	Insert a row before row containing specified cell.
	Cut	Cut to the clipboard the row containing specified cell.
	Copy	Copy to the clipboard the row containing specified cell.
	Paste Before	Paste copied or cut row before row containing specified cell.
	Paste After	Paste copied or cut row after row containing specified cell.
	Delete	Delete the row containing specified cell.

Menu	Item	Description
Cell For a command in this menu to work, click anywhere inside a cell (or explicitly select a cell or an element having a cell ancestor).	Increment Column Span	Increment the number of columns spanned by specified cell.
	Decrement Column Span	Decrement the number of columns spanned by specified cell.
	Increment Row Span	Increment the number of rows spanned by specified cell.
	Decrement Row Span	Decrement the number of rows spanned by specified cell.

Other commands:

Paste After As

The entries of this submenu allow to paste the plain text copied to the clipboard, typically using a third-party word processor or spreadsheet, as:

- one or more paragraphs,
- OR a `pre` element,
- OR one or more list items,
- OR an itemized list,
- OR one or more table rows,
- OR a table.

The last two menu entries assume that each text line specifies a table row and that, within a text line, the contents of the table cells are separated by tab characters.

Move Up

Move selected element up, that is, swap it with its preceding sibling node. Requires the element to be explicitly selected.

Move Down

Move selected element down, that is, swap it with its following sibling node. Requires the element to be explicitly selected.

Preview

Preview the document being edited using the "default viewer" helper application. The default viewer is typically a Web browser. This helper application is specified in the Preferences dialog box, Helper Applications section, Default viewer field.

2.1. Convert Document sub-menu

 This menu is present only in XMLmind XML Editor Professional Edition.

Convert to RTF (Word 2000+)

Converts the document being edited to RTF (Rich Text Format) using XMLmind FO Converter (see <http://www.xmlmind.com/foconverter/>). The document generated by this command can be edited and printed using Microsoft® Word 2000 and above.

Convert to WordprocessingML (Word 2003+)

Converts the document being edited to WordprocessingML using XMLmind FO Converter. The document generated by this command can be edited and printed using Microsoft® Word 2003 and above.

Convert to Office Open XML (Word 2007+)

Converts the document being edited to Office Open XML (.docx file) using XMLmind FO Converter. The document generated by this command can be edited and printed using Microsoft® Word 2007 and above.

Convert to OpenDocument (OpenOffice.org 2)

Converts the document being edited to OpenDocument (.odt file) using XMLmind FO Converter. The document generated by this command can be edited and printed using OpenOffice.org 2.

Print PostScript

Converts the document being edited to PostScript® using RenderX XEP (see <http://www.renderx.com/>), if its plug-in has been installed, and Apache FOP otherwise (see <http://xmlgraphics.apache.org/fop/>), and then sends the generated file to the chosen printer.

Convert to PDF

Converts the document being edited to PDF (Adobe® Portable Document Format, also known as Acrobat®) using RenderX XEP (see <http://www.renderx.com/>), if its plug-in has been installed, and Apache FOP otherwise (see <http://xmlgraphicsl.apache.org/fop/>).

All the above Convert commands display the URL chooser dialog box rather than the standard file chooser dialog box. That is, you must specify the URL (Uniform Resource Locator) of a save file.

Note that these commands can create directories on the fly, if needed to. For example, if you specify `http://www.acme.com/docs/report43/mydoc.pdf` as the URL of the save file and if directory `report43/` does not exist, this directory will be created during command execution.

3. Custom bindings

Keystroke	Action
Enter	Inside a <code>p</code> , splits the <code>p</code> in two parts. Elsewhere, normal behaviour.
Del	At the end of a <code>p</code> , if there is no node or text selection, joins this <code>p</code> with following sibling <code>p</code> if any. Otherwise, normal behaviour.
BackSpace	At the beginning of a <code>p</code> , if there is no node or text selection, joins this <code>p</code> with preceding sibling <code>p</code> if any. Otherwise, normal behaviour.
Ctrl+Enter	Inside a <code>p</code> , <code>li</code> , <code>dt</code> , <code>dd</code> , inserts same element after this one. Note that in the case of <code>dt</code> and <code>dd</code> , it inserts a new <code>dt</code> followed by a new <code>dd</code> . Elsewhere, no effect.
Shift+Ctrl+Enter	Inside a <code>p</code> , <code>li</code> , <code>dt</code> , <code>dd</code> , inserts same element before this one. Note that in the case of <code>dt</code> and <code>dd</code> , it inserts a new <code>dt</code> followed by a new <code>dd</code> . Elsewhere, no effect.
Shift+Enter	Inserts a <code>br</code> at caret and moves caret after inserted <code>br</code> .
Application Event	Action
drop	On a <code>a</code> element with attribute <code>href</code> , changes the value of the attribute to the the dropped string. Elsewhere, considers that the dropped string is a filename or an URL and therefore, attempts to open the corresponding document in the editor.

4. Table rendering

Deprecated attributes (specified in the transitional DTD) are not supported. The following attributes (specified in the strict DTD) are either completely ignored or partially supported. All other attributes (specified in the strict DTD) are supported.

Attribute	Support
table width	Ignored.
table border	Value larger than 1 treated like 1.
table cellspacing	Ignored.
table cellpadding	Ignored.
col (or colgroup) width	All forms including "20%", "3*" or "0*" are supported. A column must contain at least one cell with a column span equal to 1 for the width attribute to have an effect.
align	Values justify and char are rendered like left.
char	Ignored. See align.
charoff	Ignored. See align.

5. Spell checking

XHTML has two attributes for specifying the language of an element: lang and xml:lang. By default, XMLmind XML Editor is configured to consider xml:lang and to ignore lang. This behavior can, of course, be changed by modifying the XHTML configuration of XMLmind XML Editor (`XXE_install_dir/addon/config/xhtml/common.incl`).

A. Parameters of the XSLT stylesheets used to convert XHTML to XSL-FO

Parameter	Value	Default Value	Description
font-family	serif sans-serif	serif	The family of the font used for most body elements (paragraphs, tables, lists, etc).
font-size	Length in pt	10pt	The size of the font used for most body elements (paragraphs, tables, lists, etc). All the other font sizes are computed relatively to this font size.
hyphenate	false true	true	Specifies whether words may be hyphenated.
img-src-path	URI ending with '/'	''	By default, <code></code> is translated to <code><fo:external-graphic src='foo.gif'/></code> . If this parameter is not empty and if the value of the <code>src</code> attribute is a relative URI, then this parameter is prepended to the value of the <code>src</code> attribute. For example, this allows to generate an XSL-FO file where all the references to graphic files are absolute URIs.
page-footer-margin	Length	0.5in	The height reserved for the page footer.
page-header-margin	Length	0.5in	The height reserved for the page header.

XMLmind XML Editor - XHTML
Support

Parameter	Value	Default Value	Description
page-height	Length	Dynamic. Depends on paper-type and page-orientation.	The height of the printed page.
page-margin-bottom	Length	1in	The distance between the bottom of the page and the bottom of the printed area.
page-margin-left	Length	1in	The distance between the left of the page and the left of the printed area.
page-margin-right	Length	1in	The distance between the right of the page and the right of the printed area.
page-margin-top	Length	1in	The distance between the top of the page and the top of the printed area.
page-number-print-in-footer	false true	true	Specifies whether the page footer should include the page number.
page-orientation	portrait landscape	portrait	The orientation of the printed page .
page-width	Length	Dynamic. Depends on paper-type and page-orientation.	The width of the printed page.
paper-type	2A0 4A0 A0 A1 A10 A2 A3 A4 A5 A6 A7 A8 A9 B0 B1 B10 B2 B3 B4 B5 B6 B7 B8 B9 C0 C1 C10 C2 C3 C4 C5 C6 C7 C8 C9 USletter	USletter	A convenient way to specify the size of the printed page. It is also possible to specify a custom paper type by ignoring the <code>paper-type</code> parameter and directly specifying the <code>page-width</code> and <code>page-height</code> parameters.
root-id	ID	''	If this parameter is not empty, it must be the value of an <code>id</code> attribute that occurs in the document being formatted. The entire document will be loaded, but formatting will begin at the element identified, rather than at the root element. For example, this allows to convert a specific <code>div</code> element rather than the whole XHTML document.
text-align	justify start	justify	Specifies whether text (e.g. in paragraphs) should be justified (that is, flush left and right) or just left aligned (that is, flush left and ragged right).
title-print-in-header	false true	true	Specifies whether the page header should include the title of the XHTML document being converted.