XMLmind XML Editor - XHTML Support

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

April 2, 2013

Table of Contents

1. Creating an XHTML document

The File → New dialog box allows you to choose between the following document templates:

XHTML Version	Template Name	Description
1.0	XHTML Page (Strict)	File having a `.xhtml` suffix, starting with an XML declaration (`<?xml version="1.0"?>`) and a `<!DOCTYPE>` pointing to the XHTML 1.0 Strict DTD.
	HTML Page (Strict)	File having a `.html` suffix, having no XML declaration[a] and starting with a `<!DOCTYPE>` pointing to the XHTML 1.0 Strict DTD. This document template is a well-formed, valid XHTML file which is intended to be seen by Web browsers as an HTML file.
	XHTML Page (Transitional)	File having a `.xhtml` suffix, starting with an XML declaration and a `<!DOCTYPE>` pointing to the XHTML 1.0 Transitional DTD.
	HTML Page (Transitional)	File having a `.html` suffix, having no XML declaration and starting with a `<!DOCTYPE>` pointing to the XHTML 1.0 Transitional DTD. This document template is a well-formed, valid XHTML file which is intended to be seen by Web browsers as an HTML file.
1.1	XHTML Page	File having a `.xhtml` suffix, starting with an XML declaration and a `<!DOCTYPE>` pointing to the XHTML 1.1 DTD.
	HTML Page	File having a `.html` suffix, having no XML declaration and starting with a `<!DOCTYPE>` pointing to the XHTML 1.1 DTD. This document template is a well-formed, valid XHTML file which is intended to be seen by Web browsers as an HTML file.
5.0	XHTML Page	File having a `.xhtml` suffix, starting with an XML declaration followed by `<!DOCTYPE html>` and conforming to an XHTML 5 W3C XML Schema developed by XMLmind.

XHTML Version	Template Name	Description
	HTML Page	File having a `.html` suffix, having no XML declaration, starting with `<!DOCTYPE html>` and conforming to an XHTML 5 W3C XML Schema developed by XMLmind. This document template is a well-formed, valid XHTML file which is intended to be seen by Web browsers as an HTML file.

[a]This implies that the encoding of the file must be UTF-8 for this file to be successfully opened in an XML editor.

This is enforced by the fact that all document templates called "HTML Page" contain:

```
<meta content="text/html; charset=UTF-8"
      http-equiv="Content-Type" />
```

All the above document templates should be highly interchangeable between Web browsers and XML editors, however it is worth keeping in mind the following points:

- If you want to feed your XHTML document directly to a Web browser, choose a document template called "HTML Page".
- If you want to feed your XHTML document directly to a Web browser but your document also contains MathML, choose a document template called "XHTML Page".
- If your XHTML document is a building block which is intended to be processed by an XML based publishing system, choose a document template called "XHTML Page".
- More information in *Activating Browser Modes with Doctype*.

2. Opening an existing XHTML document

- If your document starts with `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN">`, it will be detected as being an XHTML document associated with the XMLmind XML Editor (XXE for short) configuration called "XHTML Strict".

  Note that the XHTML 1.0 Strict DTD does not allow the `body` element to contain text other than whitespace.

- If your document starts with `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">`, it will be detected as being an XHTML document associated with the XXE configuration called "XHTML Transitional".

- If your document starts with `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN">`, it will be detected as being an XHTML document associated with the XXE configuration called "XHTML 1.1".

- If your document does *not* start with `<!DOCTYPE>` and if its root element is html in the "http://www.w3.org/1999/xhtml" namespace, it will be detected as being an XHTML document associated with the XXE configuration called "XHTML 5.0".

- Otherwise your document will not be detected as being an XHTML document and you'll have to customize one of the stock XHTML configurations if you want to open your document as an XHTML one.

3. CSS stylesheets found in the View menu

The View menu allows to choose between two CSS stylesheets:

Semantic
: This stylesheet contains styles which are intended to reflect the purpose of each XHTML element. This stylesheet makes it easy editing XHTML documents in a validating XML Editor such as XXE. This is the default CSS stylesheet.

  Note that the following elements are not styled at all, that is, they are rendered as tree views embedded in the styled view: `base`, `meta`, `link`, `style`, `script`, `iframe`, `embed`, `object`, `param`, `video`, `audio`, `source`, `track`,

`canvas`, `map`, `area`, `input`, `button`, `select`, `datalist`, `optgroup`, `option`, `textarea`, `keygen`, `output`, `progress`, `meter`, `command`.

Emulate Web Browser

This stylesheet emulates (to a certain extent) how a Web browser typically renders an HTML page. When this stylesheet has been selected, XXE will dynamically apply all the CSS styles found in `style` attributes, `style` elements and `link` elements pointing to CSS stylesheets.

Sometimes XXE will fail to detect a change in how the XHTML document being edited should be styled. In such case, you'll have to reload the "Emulate Web Browser" stylesheet by reselecting this item in the View menu. For example, the following changes will *not* automatically trigger style changes:

- Modify the textual content of a `style` element.
- Paste a `div` element containing a `style` element having no `scoped` attribute.

Note

When XXE finds errors in the CSS styles it attempts to apply dynamically, it will not report these errors. Instead, it will log them in a special log called "User Styles". If you suspect that XXE has found errors in your CSS styles, you may want to view the content of this log by clicking the Show Message Log button in *XMLmind XML Editor - Online Help* which is next to the status line found at the bottom of XXE's window[1] and then select "User Styles" from the Category combobox.

Tip

If you often use `<div class="XXX">` and/or `<span class="YYY">` elements in your XHTML documents and want to see these elements properly styled using the `<link rel="stylesheet">` element found in the `head` of your XHTML documents, you may want to make the "Emulate Web Browser" stylesheet your default one. In order to do this, simply select this stylesheet from the View menu and then select Options → Customize Configuration → Save Views As Default[2].

4. The XHTML menu

Table editing commands fully support HTML 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.
	Sort Rows	Sort all the rows of the table according to the string values of the cells of the “selected column”. (The “selected column” is the column containing specified cell.) A dialog box is displayed allowing to specify the following sort options:

[1] Or simply right-click on the status line.

[2] If you are a consultant customizing XXE for a group of writers, you'll have to customize the XHTML configuration by adding to it a `windowLayout` configuration element in *XMLmind XML Editor - Configuration and Deployment*.

Menu	Item	Description
		Order Dictionary is the language-specific alphabetical order. Example: (`Charles`, `best`, `Albert`) is sorted as (`Albert`, `best`, `Charles`). Numeric. The string value of a cell is expected to start with a number. Example: (`+15.0%`, `1.50%`, `-20%`) is sorted as (`-20%`, `1.50%`, `+15.0%`). Lexicographic is the order of Unicode characters. Example: (`Charles`, `best`, `Albert`) is sorted as (`Albert`, `Charles`, `best`). Dictionary and Numeric orders will cause this menu item to fail, unless the language of the table can be determined (i.e. lookup for the `lang` or `xml:lang` attribute). Direction Ascending means: A to Z, low to high. Descending means: Z to A, high to low. Note that: • Header/footer rows (i.e. `thead`) are never sorted. • The contents of row groups (i.e. `tbody`) are sorted separately.
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.
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 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.

Tip

If you need to paste the copied text as an ordered list, first paste this text as an itemized list then convert the pasted list to an ordered list using Edit → Convert (**Ctrl**+**T**).

The above menu entries paste elements *after* the implicitly or explicitly selected element.

The following entries of this submenu allow to paste the *image* copied to the clipboard as:
- `img`,
- `div` (containing an `img`)
- `figure` (XHTML5 only).

Except for menu entry "img" which replaces the selection or pastes an element at caret position (like Edit → Paste), the other menu entries paste an element *after* the implicitly or explicitly selected element.

 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.

4.1. Convert Document sub-menu

Convert to RTF (Word 2000+)
Converts the document being edited to RTF (Rich Text Format) using XMLmind XSL-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 XSL-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 XSL-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 XSL-FO Converter. The document generated by this command can be edited and printed using OpenOffice.org 2 and above.

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://xmgraphicsl.apache.org/fop/).

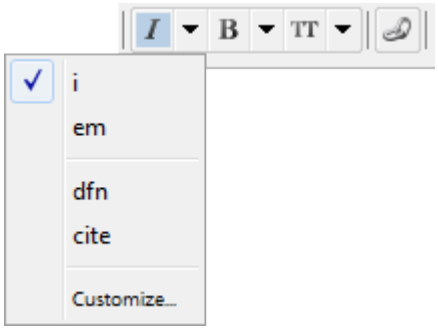
All the above Convert commands display the URL chooser dialog box rather than the standard file chooser dialog box. That is, you must specify the URL (Uniform Resource Locator) of a save file.

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.

5. The XHTML toolbar

The XHTML tool bar starts with a number of “text style” toggles. These toggles emulate the behavior of the Bold, Italic, Underline, etc, toggles found in the tool bars of almost all word-processors. More information about text style toggles in About “text style” toggles in *XMLmind XML Editor - Online Help*.

Figure 1. Toggles found at the beginning of the XHTML tool bar



In the above screenshot, the caret is inside an `i` element and the user clicked the arrow button next to a “italic text style” toggle.

I Toggle i
: “Toggle” element `i`. Next to this toggle is found an arrow button displaying a menu containing additional checkboxes for the following elements: `em`, `dfn`, `cite`.

B Toggle b
: “Toggle” element `b`. Next to this toggle is found an arrow button displaying a menu containing additional checkboxes for the following elements: `strong`, `abbr`.

TT Toggle tt
: “Toggle” element `tt`. Next to this toggle is found an arrow button displaying a menu containing additional checkboxes for the following elements: `code`, `var`, `kdb`, `samp`.

Toggle a[href]
: “Toggle” hypertext link, that is, an a element having an `href` attribute.

¶ Add p
: Add a `p` element after node selection or after caret at a location where it is valid to do so.

Add li or dt+dd
Add a `li` element or a `dt`/`dd` pair after current list item. For this command to work, suffice to click anywhere inside an `ul`, `ol` or `dl` element.

Add ul
Add an `ul` element after node selection or after caret at a location where it is valid to do so.

Add ol
Add an `ol` element after node selection or after caret at a location where it is valid to do so.

Add dl
Add an `dl` element after node selection or after caret at a location where it is valid to do so.

Add pre
Add an `pr` element after node selection or after caret at a location where it is valid to do so.

Add table
Displays a menu which allows to add several styles of tables after node selection or after caret at a location where it is valid to do so.

Add img
Displays a menu which allows to

- insert an `img` element at caret position;
- OR add a `div` element containing an `img` or a `figure` element containing an `img` (XHTML 5 only) after node selection or after caret at a location where it is valid to do so.

Add heading
Displays a menu which allows to add an `h1`, `h2`, `h3`, etc, element after node selection or after caret at a location where it is valid to do so.

Add section
(XHTML 5 only.) Displays a menu which allows to add a `section`, `aside`, `article`, `header`, `footer` or `hgroup` element after node selection or after caret at a location where it is valid to do so.

6. Custom bindings

Keystroke	Action
Enter	Insert a newline character if possible. Otherwise, if caret is at the beginning of a paragraph, list item or a few other kinds of block, insert same block before. Otherwise, if caret is at the end of a block, insert same block after. Otherwise, split block.
Del	Delete selection if any. Otherwise, if caret is at the end of a paragraph, list item or a few other kinds of block, join with following block. Otherwise, delete character following caret.
BackSpace	Delete selection if any. Otherwise, if caret is at the beginning of a paragraph, list item or a few other kinds of block, join with preceding block. Otherwise, delete character preceding caret.

Keystroke	Action
Ctrl+**Enter**	Add same block after the paragraph, list item or a few other kinds of block which is the ancestor of selected node.
Shift+**Ctrl**+**Enter**	Add same block before the paragraph, list item or a few other kinds of block which is the ancestor of selected node.
Shift+**Enter**	Inserts a `br` at caret and moves caret after inserted `br`.
Application Event	**Action**

Drop an object.	• On a `a` element having an `href attribute`, change the value of this attribute to the dropped string.
	• On an `img` element, change the value of the `src` attribute to the dropped string.
	• Elsewhere
	  • If the object being dropped represents an URL or an absolute filename, open the corresponding document in XMLmind XML Editor.
	  • Otherwise, paste the dropped text or XML at or after the drop location.

Drag one of the “handles” displayed around an image. (The “handles” are displayed after clicking on the image.)	Resize the image, but always preserve its aspect ratio. Pressing Ctrl (Cmd on the Mac) while dragging the handle allows to distort the image.
Drag a separator found between two table columns.	Resize the table column. More precisely, this gives an appropriate percent width (e.g. `<colgroup width="35%">`) to *all* table columns. **Warning** If the document containing the table is an XHTML 5 document, then the `colgroup` element has no `width` attribute. In such case, the resize table column action has to use the `style` attribute (e.g. `<colgroup style="width: 35%">`). In practice, this means that unless you select the "Emulate Web Browser" CSS stylesheet [3], you'll not see the table columns correctly resized.

A. Table rendering

Deprecated attributes (that is, those specified in the transitional DTD 1.0) are not supported. The following attributes are either completely ignored or partially supported. All the other attributes 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.

Attribute	Support
	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`.

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

Parameter	Value	Default Value	Description
`apply-css-styles`	`'no'` | `'yes'`	`'yes'`	Specifies whether CSS styles specified in XHTML `style` attributes, `style` and `link` elements also apply to the XSL-FO file[a]. Depending on the context, the following CSS properties are converted to their equivalent XSL-FO attributes. The corresponding shorthand CSS properties are supported too. Any other CSS property is ignored. • margin-top, margin-right, margin-bottom, margin-left. • padding-top, padding-right, padding-bottom, padding-left. • border-top-style, border-right-style, border-bottom-style, border-left-style. • border-top-width, border-right-width, border-bottom-width, border-left-width. • border-top-color, border-right-color, border-bottom-color, border-left-color. • background-color, background-image, background-repeat, background-position. • color. • font-family, font-style, font-weight, font-size. • text-decoration. • text-align. • text-indent. • vertical-align. • line-height. • list-style-type, list-style-position, list-style-image. • width, height. • caption-side. • border-spacing. **Important** This feature, which leverages XMLmind XML Editor CSS engine, will not work unless the XSLT stylesheets are being invoked by an XMLmind application such as XMLmind XML Editor or XMLmind XSL Utility.
`base-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.

Parameter	Value	Default Value	Description
`external-href-after`	String	`']'`	Appended after the external URL referenced by an a element. Ignored unless `show-external-links='yes'` [11].
`external-href-before`	String	`' ['`	Separates the text of an a element from its referenced external URL. Ignored unless `show-external-links='yes'` [11].
`font-family`	One or more font families separated by commas	`'serif'`	The font family used by default for all elements.
`footer-center`	A mix of text and variables.	`''`	Specifies the contents of the central part of a page footer. See the section called “Specifying a header or a footer” [12].
`footer-center-width`	String representing an integer larger than or equal to 1.	`'6'`	Specifies the proportional width of the central part of a page footer. See the section called “Specifying a header or a footer” [12].
`footer-left`	A mix of text and variables.	`'{{even-page-number}}'`	Specifies the contents of the left part of a page footer. See the section called “Specifying a header or a footer” [12].
`footer-left-width`	String representing an integer larger than or equal to 1.	`'2'`	Specifies the proportional width of the left part of a page footer. See the section called “Specifying a header or a footer” [12].
`footer-right`	A mix of text and variables.	`'{{odd-page-number}}'`	Specifies the contents of the right part of a page footer. See the section called “Specifying a header or a footer” [12].
`footer-right-width`	String representing an integer larger than or equal to 1.	`'2'`	Specifies the proportional width of the right part of a page footer. See the section called “Specifying a header or a footer” [12].
`footer-separator`	`'no'` | `'yes'`	`'yes'`	Specifies whether an horizontal rule should be drawn above the page footer. See the section called “Specifying a header or a footer” [12].
`header-center`	A mix of text and variables.	`'{{document-title}}'`	Specifies the contents of the central part of a page header. See the section called “Specifying a header or a footer” [12].
`header-center-width`	String representing an integer larger than or equal to 1.	`'6'`	Specifies the proportional width of the central part of a page header. See the section called “Specifying a header or a footer” [12].
`header-left`	A mix of text and variables.	`''`	Specifies the contents of the left part of a page header. See the section called “Specifying a header or a footer” [12].
`header-left-width`	String representing an integer larger than or equal to 1.	`'2'`	Specifies the proportional width of the left part of a page header. See the section called “Specifying a header or a footer” [12].
`header-right`	A mix of text and variables.	`''`	Specifies the contents of the right part of a page header. See the section called “Specifying a header or a footer” [12].

Parameter	Value	Default Value	Description
header-right-width	String representing an integer larger than or equal to 1.	'2'	Specifies the proportional width of the right part of a page header. See the section called “Specifying a header or a footer” [12].
header-separator	'no' | 'yes'	'yes'	Specifies whether an horizontal rule should be drawn below the page header.
hyphenate	'no' | 'yes'	'no'	Specifies whether words may be hyphenated.
justified	'no' | 'yes'	'no'	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).
page-orientation	'portrait' | 'landscape'	'portrait'	The orientation of the printed page.
page-ref-after	String	']'	Appended after the page number pointed to by an a element. Ignored unless show-xref-page='yes' [11].
page-ref-before	String	' ['	Separates the text of an a element from the page number it points to. Ignored unless show-xref-page='yes' [11].
paper-type	Allowed values are: 'Letter', 'Legal', 'Ledger', 'Tabloid', 'A0', 'A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10', 'B0', 'B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B9', 'B10', 'C0', 'C1', 'C2', 'C3', 'C4', 'C5', 'C6', 'C7', 'C8', 'C9', 'C10' *(case-insensitive)*.	'A4'	A convenient way to specify the size of the printed page. It is also possible to specify a custom paper type by ignoring the paper-type parameter and directly specifying the page-width [14] and page-height [13] parameters.
show-external-links	'no' | 'yes'	'no'	Specifies whether the external URL referenced by an a element should be displayed right after the text contained by this element. Example: show-external-links='yes' causes Oasis to be rendered as follows: Oasis [http://www.oasis-open.org/].
show-xref-page	'no' | 'yes'	'no'	Specifies whether the page number corresponding to the internal link target referenced by an a element should be displayed right after the text contained by this element. Example: show-xref-page='yes' causes Introduction to be rendered as follows: Oasis [3].
two-sided	'no' | 'yes'	'no'	Specifies whether the document should be printed double sided.

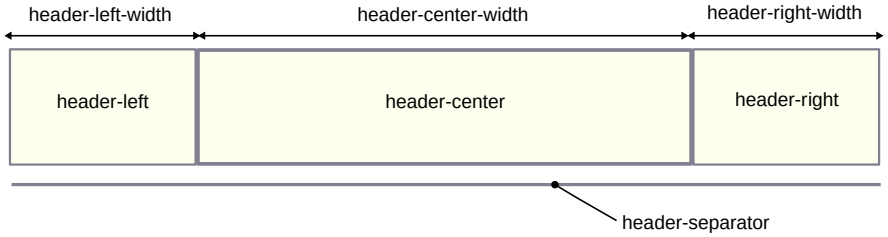
Parameter	Value	Default Value	Description
`ul-li-bullets`	One or more bullet characters separated by spaces	`' &#x2022; &#x2013;'`	Specify which bullet character to use for an `ul`/`li` element. Additional characters are used for nested `li` elements. For example, if `ul-li-bullets="* - +"`, "`*`" will be used for `ul`/`li` elements, "`-`" will be used for `ul`/`li` elements contained in a `ul`/`li` element and "`+`" will be used for `ul`/`li` elements nested in two `ul`/`li` elements.

[a]In addition to the presentation attributes (`xsl:attribute-set`) specified in the XSLT stylesheets that generate the XSL-FO file

Specifying a header or a footer

The header or the footer of a generated PDF, RTF, etc, page has 3 columns.

Figure B.1. Layout of a header



The width of these columns may be specified using the `header-left-width` [10], `header-center-width` [10], `header-right-width` [11] parameters for the header and the `footer-left-width` [10], `footer-center-width` [10], `footer-right-width` [10] parameters for the footer.

The width of a column is specified as an integer which is larger than or equal to 1. This value is the *proportional width* of the column. For example, if the left column has a width equal to 2 and the right column has a width equal to 4, this simply means that the right column is twice (4/2 = 2) as wide as the left column.

The contents of these columns may be specified using the `header-left` [10], `header-center` [10], `header-right` [10] parameters for the header and the `footer-left` [10], `footer-center` [10], `footer-right` [10] parameters for the footer.

When `header-left`, `header-center`, `header-right` are all specified as the empty string, no header is generated. When `footer-left`, `footer-center`, `footer-right` are all specified as the empty string, no footer is generated.

The content of a column is a mix of text and variables. Example: "`Page {{page-number}} of {{page-count}}`".

Supported variables are:

`{{document-title}}`
: The title of the document.

`{{current-heading}}`
: In principle, the title of the last `h1`, `h2`, `h3`, etc, element being rendered.

  No effect unless the XSLT stylesheets are customized in order to give a value to this variable.

```
<fo:marker
  marker-class-name="current-heading">
  ...
</fo:marker>
```

`{{page-number}}`
: Current page number.

`{{odd-page-number}}`
: Current page number; empty when a double-sided output is being generated and the current page number is even.

`{{even-page-number}}`
: Current page number when a double-sided output is being generated and the current page number is even; empty otherwise.

`{{page-count}}`
: Total number of pages of the document.

`{{break}}`
: A line break.

`{{image(`*`URI`*`)}}`
: An image having specified URI. A relative URI is resolved against the current working directory. Example: "`{{image(artwork/logo.svg)}}`".

  The image is displayed at its intrinsic size. If it is too tall to fit in a page header or footer, you may need to increase the height of the header or footer by using the following parameters: `header-height` [13], `body-top-margin` [13], `footer-height` [13], `body-bottom-margin` [13].

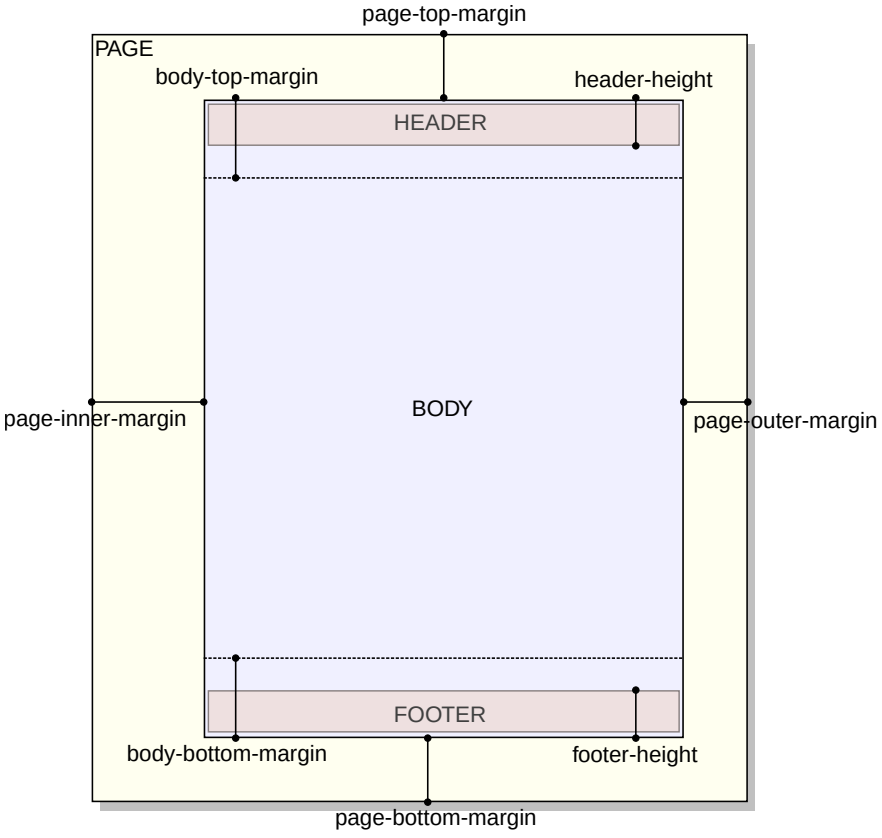
`{{url(`*`URI`*`)}}`
: A styled hyperlink to specified URI. No attempt is made to resolve a relative URI. The specified URI may be optionally followed by a space character and then, the text of the link. Examples: "`{{url(http://www.xmlmind.com/xmleditor/)}}`", "`{{url(http://www.xmlmind.com/xmleditor/ XMLmind XML Editor)}}`"

Advanced parameters

Parameter	Value	Default Value	Description
`body-bottom-margin`	Length	`'0.5in'`	See figure [14] below.
`body-top-margin`	Length	`'0.5in'`	See figure [14] below.
`footer-height`	Length	`'0.4in'`	See figure [14] below.
`header-height`	Length	`'0.4in'`	See figure [14] below.
`img-src-path`	URI ending with `'/'`	`''`	If this parameter is not empty and if the value of the `src` attribute is a relative URI, then this parameter is prepended to the value of the `src` attribute. This low-level alternative to `resolve-img-src='yes'` [14] also allows to generate an XSL-FO file where all the references to graphic files are absolute URIs.
`page-bottom-margin`	Length	`'0.5in'`	See figure [14] below.
`page-height`	Length. Example: `'297mm'`.	Depends on `paper-type`.	The height of the printed page.
`page-inner-margin`	Length	if parameter `two-sided` is specified as `'yes'` then `'1.25in'` otherwise `'1in'`.	See figure [14] below.
`page-outer-margin`	Length	Default value: if parameter `two-sided` is spe-	See figure [14] below.

Parameter	Value	Default Value	Description
		cified as `'yes'` then `'0.75in'` otherwise `'1in'`.	
`page-top-margin`	Length	`'0.5in'`	See figure [14] below.
`page-width`	Length. Example: `'8.5in'`.	Depends on `paper-type`.	The width of the printed page.
`resolve-a-href`	`'no'` | `'yes'`	`'no'`	In the XSL-FO file, convert relative URIs contained in the `href` attribute of `a` elements to absolute URIs. This is done by resolving the relative URI against the base of the `a` element.
`resolve-img-src`	`'no'` | `'yes'`	`'yes'`	In the XSL-FO file, convert relative URIs contained in the `src` attribute of `img` elements to absolute URIs. This is done by resolving the relative URI against the base of the `img` element.
`root-id`	ID found in the source XHTML document	`''`	If this parameter is not empty, it must be the value of an `id` 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 `div` element rather than the whole XHTML document.
`screen-resolution`	Number	`96.0`	Screen resolution in DPI. Used to convert `px` to `pt`.

Figure B.2. Page areas



System parameters

Note

The following parameters are expected to be automatically specified by the application invoking the XSLT stylesheets. Such system parameters are not intended to be specified by the end-user. Such system parameters are documented here only because the end-user may see them referenced in some dialog boxes or in some configuration files.

Parameter	Value	Default Value	Description
`foProcessor`	String. Examples: '`FOP`', '`XEP`', '`XFC`'.	If `outputFormat` [15] has been specified and if system property XSL_FO_PROCESSORS contains the list of all available XSL-FO processors, the default value is computed, otherwise the default value is `''`.	The name of the XSL-FO processor used to convert the XSL-FO file generated by the XSLT stylesheets to the target output format.
`outputFormat`	String. Examples: '`ps`', '`pdf`', '`rtf`', '`wml`', '`docx`', '`odt`'.	`''`	The file extension of the target output file.