
XMLmind XML Editor - XHTML Support

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

February 22, 2019

Abstract

This document, which is also available using the online help system of XXE, describes the commands which are specific to XHTML.

Table of Contents

1. Creating an XHTML document	1
2. Opening an existing XHTML document	2
3. CSS stylesheets found in the View menu	2
3.1. Ruby, form controls and embedded content	3
4. The XHTML menu	4
4.1. Convert Document sub-menu	5
5. The XHTML toolbar	6
5.1. Table editor	9
5.2. The <code>iframe</code> editor dialog box	10
6. Custom bindings	13
A. Table rendering	14
B. Parameters of the XSLT stylesheets used to convert XHTML to XSL-FO	15

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 <code>.xhtml</code> suffix, starting with an XML declaration (<code><?xml version="1.0" ?></code>) and a <code><!DOCTYPE></code> pointing to the XHTML 1.0 Strict DTD.
	HTML Page (Strict)	File having a <code>.html</code> suffix, having no XML declaration ^a and starting with a <code><!DOCTYPE></code> 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 <code>.xhtml</code> suffix, starting with an XML declaration and a <code><!DOCTYPE></code> pointing to the XHTML 1.0 Transitional DTD.
	HTML Page (Transitional)	File having a <code>.html</code> suffix, having no XML declaration and starting with a <code><!DOCTYPE></code> 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 <code>.xhtml</code> suffix, starting with an XML declaration and a <code><!DOCTYPE></code> pointing to the XHTML 1.1 DTD.
	HTML Page	File having a <code>.html</code> suffix, having no XML declaration and starting with a <code><!DOCTYPE></code> pointing to the XHTML 1.1 DTD.

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

^aThis 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".
- 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 the following two CSS stylesheets:

Semantic (default CSS stylesheet)

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.

Note that in the "XHTML 5" configuration, the "Semantic" stylesheet dynamically applies the `style` attribute to table elements such as `table`, `colgroup`, `tbody`, `td`, etc¹. This counterbalances the fact that, in XHTML 5, all attributes influencing table rendering (`frame`, `rules`, `width`, `valign`, etc) have been removed from table elements.

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 window² and then select "User Styles" from the Category combobox.

Tip

If you often use `<div class="XXX">` and/or `` 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 Default³.

3.1. Ruby, form controls and embedded content

Both the above CSS stylesheets support ruby, form controls and embedded content (`iframe`, `object`, `video`, etc). However in the case of the `input` form control, some attributes such as `autocomplete`, `list`, `maxlength`, `minlength`, etc, are not yet supported.

The "Emulate Web Browser" stylesheet renders the following elements the same way as in a Web browser, at the expense of a less convenient editing: `ruby`, the `select` form control and most embedded content elements.

A demo document showing these features is available in `xxe_install_dir/demo/xhtml15-objects.html`.

¹Only the `style` attribute and only the table elements, unlike the "Emulate Web Browser" stylesheet.

²Or simply right-click on the status line.

³If you are a consultant customizing XXE for a group of writers, you'll have to customize the XHTML configuration by adding to it a `window-Layout` configuration element in *XMLmind XML Editor - Configuration and Deployment*.

4. The XHTML menu

Paste As

Paste from Word

Pastes “rich text” copied to the clipboard using MS-Word 2003+.

The pasted data replaces the text or node selection if any. When there is no selection, XMLmind XML Editor automatically determines a valid insertion location at or following the caret position.

If XMLmind XML Editor fails to find such valid insertion location, the rich text is converted to valid XHTML and then copied to the clipboard, overwriting the original data put there by MS-Word. This allows to use the “normal” Paste Before, Paste or Paste After commands to paste the data elsewhere in the document.

How to import an entire MS-Word document as XHTML

1. Open the document in MS-Word.
2. Press **Ctrl+A** (Select All) then press **Ctrl+C** (Copy) to copy it to the clipboard.
3. Create a new XHTML document in XME by using File → New.
4. Use File → Save As to save this new XHTML document to disk.
5. Explicitly select the root element of the XHTML document, for example by clicking on its name in the node path bar.
6. Select menu item "Paste from Word" to paste the content of the clipboard⁴.

Tip

If, using MS-Word, you want to copy a piece of text rather than a paragraph, do not include the hidden character found at the very end of a paragraph (the *paragraph mark*) in your selection.

This menu item is available only on Windows and on the Mac. On Linux, or more simply if you don't need this feature, you may uninstall it using Options → Install Add-ons, and then selecting the add-on called "Paste from Word".

Note

If you are not satisfied with the result of "Paste from Word", please be kind enough to send your `.doc` or `.docx` file to `<xmleditor-info@xmlmind.com>` (unlike `<xmleditor-support@xmlmind.com>`, this email address is *not* a public mailing list). Please understand that collecting as many difficult cases as possible is absolutely needed to improve this feature.

Other menu entries

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.

⁴Note that **Ctrl+V**, that is, the plain Edit → Paste command, would not work here.

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 following entries of this submenu allow to paste the *image* copied to the clipboard as:

- `img`,
- `div` (containing an `img`)
- `figure` (XHTML5 only).

Menu entry "img" replaces the text or node selection if any. When there is no selection, this menu entry pastes its element at caret position (just like Edit → Paste).

All the other menu entries also replace the text or node selection if any. When there is no selection, these menu entries paste their elements at any valid position in the document following the caret position.

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.

Note that this command automatically creates a *temporary preview file* in any of the following cases:

- The XHTML document to be previewed contains XIncludes. XIncludes are transcluded in the temporary preview file.
- Save option `favorInteroperability` in *XMLmind XML Editor - Configuration and Deployment* is turned on, which is the case by default, and the XHTML document to be previewed contains processing instructions. Processing instructions are removed from temporary preview file.

Tip

If you want to preview the document being edited as is, do not use XHTML → Preview. Instead please use Tools → Helper Application → Open Document in Helper Application.

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.

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/>).

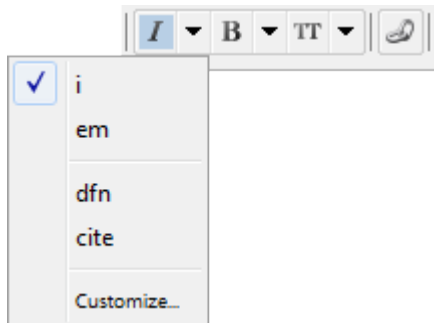
All the above Convert commands display the URL chooser dialog box rather than the standard file chooser dialog box. That is, you must specify the URL (Uniform Resource Locator) of a save file.

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`.

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.

Convert to plain text

Convert implicit or explicit selection to plain text.

Add p

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

Decrease nesting level

Convert a paragraph to a list item and a list item to a paragraph, the new element having a lesser nesting level than the original one. This button automatically splits lists when needed to.

This is the inverse command of "Increase nesting level". More Information below [7].

Increase nesting level

Convert a paragraph to a list item and a list item to a paragraph, the new element having a greater nesting level than the original one. This button automatically creates lists or merges adjacent lists when needed to.

Note that the two above buttons strictly alternate between paragraphs and list items. This means that you'll often have to click a button *twice in a row*. For example, in order to create a nested list, first click anywhere inside a list item and then click "Increase nesting level" twice. First click converts the list item to a plain paragraph contained in the preceding list item. Second click converts this paragraph to the first item of a new nested list.

For the two above buttons to function, any of the following conditions should be met:

- A sequence of list items must be explicitly selected.
- A list must be explicitly selected. This is equivalent to selecting all its items.
- A sequence of blocks *starting with a paragraph* must be explicitly selected.
- A paragraph must be implicitly selected. In order to implicitly select a paragraph, suffice to click anywhere inside it. However if this paragraph is the first child of a list item, then it's the list item which is implicitly selected.
- A list item to be implicitly selected. In order to implicitly select a list item, suffice to click anywhere inside it.


Change list type

Displays a popup menu allowing to change the type of the current list.

The list must be explicitly or implicitly selected. In order to implicitly select a list, suffice to click anywhere inside it.

Add list

Displays a menu allowing to select a type of list (`ul`, `ol`, `dl`). The chosen list is added 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 pre

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

Menu entry "Normalize Whitespace" normalizes whitespace in implicitly or explicitly selected `pre` element. Normalizing whitespace means: expanding tab characters to a number of space characters and removing the space characters which are common to the beginning of all text lines (that is, removing the superfluous "indentation" in the `pre` element, if any).

 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.

 Table editor


See Section 5.1, "Table editor" [9].

 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;
- OR add a `div` element containing an `img` and a `map` after node selection or after caret at a location where it is valid to do so.

After using this "img+map" menu item, right-click anywhere inside the newly inserted `map` element and select "Edit Image Map" from the contextual popup menu to display an *image map editor*. This image map editor allows to add "hot areas" to your image. More information in Section 18, "The "Edit Image Map" dialog box" in *XMLmind XML Editor - Online Help*.

 Insert media object

This toolbar button is present only when an XHTML 5 document is opened. Displays a menu allowing to insert a media element at caret position.

audio

XHTML 5 `audio` element allowing to add audio to your document. More information in Section 17, "The media player dialog box" in *XMLmind XML Editor - Online Help*.

video

XHTML 5 `video` element allowing to add video to your document. More information in Section 17, "The media player dialog box" in *XMLmind XML Editor - Online Help*.

iframe(youtube)

An `iframe` element showing a sample YouTube™ video. More information in Section 5.2, "The `iframe` editor dialog box" [10].

H1 ▾ 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` or `footer` element after node selection or after caret at a location where it is valid to do so.

5.1. Table editor

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

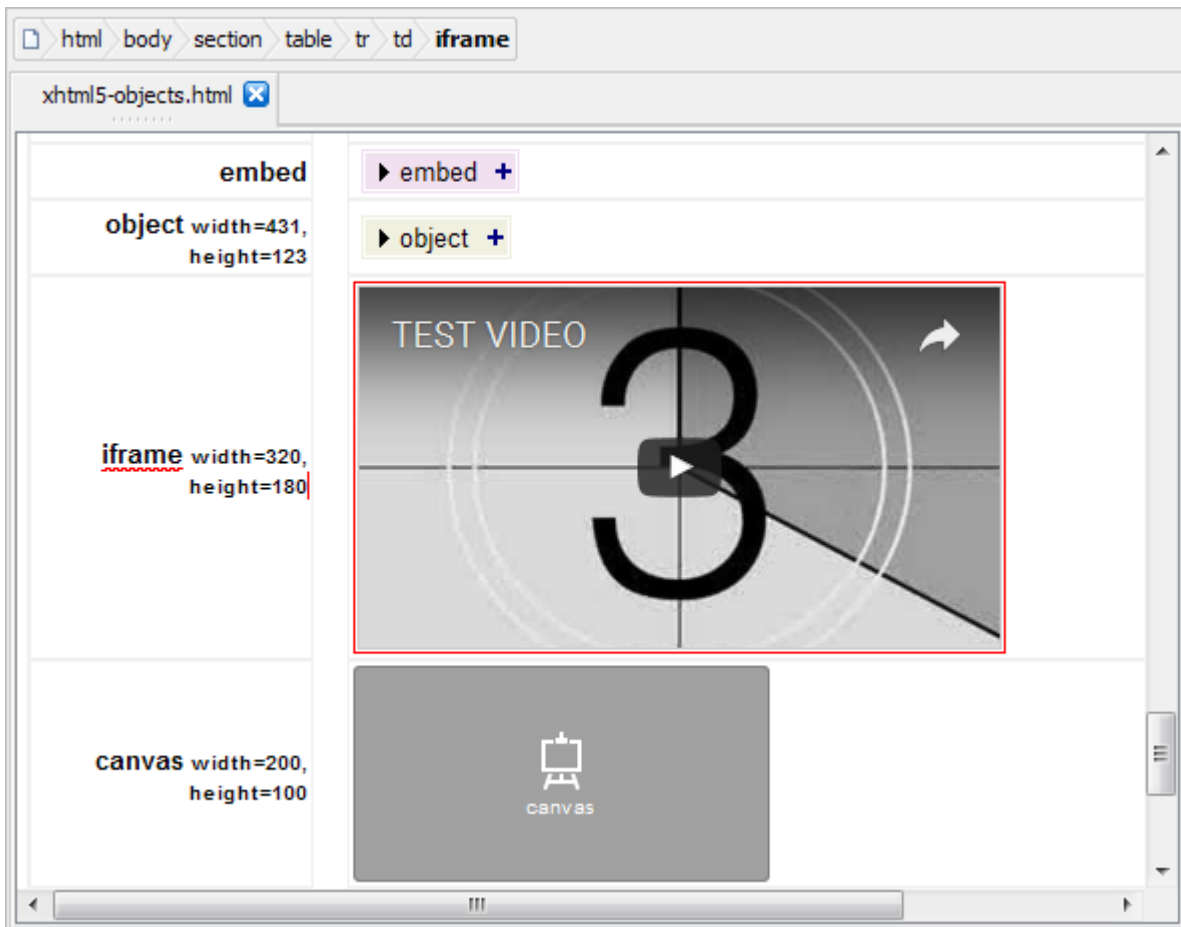
Button	Menu item	Description
 Table column For a command in this menu to work, click anywhere inside a cell (or explicitly select a cell or an element having a cell ancestor).	 Insert Before	Insert a column before column containing specified cell.
	 Insert After	Insert a column after column containing 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: 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 <code>lang</code> or <code>xml:lang</code> attribute). Direction Ascending means: A to Z, low to high. Descending means: Z to A, high to low. Note that: <ul style="list-style-type: none"> • Header/footer rows (i.e. <code>thead</code>) are never sorted. • The contents of row groups (i.e. <code>tbody</code>) are sorted separately.

Button	Menu item	Description
 Table 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. <p style="text-align: center;">Note</p> 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.
 Table 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.
	 Set Color (XHTML 1.0 and 1.1 configurations)	Displays a dialog box allowing to give a background color to specified cell. Unlike the other entries of this menu, this entry allows to give a background color, not only to specified cell, but also to one or more of any of the following explicitly selected elements: <code>table</code> , <code>colgroup</code> , <code>col</code> , <code>thead</code> , <code>tfoot</code> , <code>tbody</code> , <code>tr</code> , <code>td</code> , <code>th</code> .
	 Format (XHTML 5 configuration)	Displays a dialog box allowing to format specified cell. This dialog box makes it easy and safe setting the <code>style</code> attribute of specified cell to one or more of the following CSS properties: <code>text-align</code> , <code>vertical-align</code> , <code>background-color</code> , <code>border-style</code> , <code>border-width</code> , <code>border-color</code> . Unlike the other entries of this menu, this entry allows to format, not only to specified cell, but also one or more of any of the following explicitly selected elements: <code>table</code> , <code>colgroup</code> , <code>col</code> , <code>thead</code> , <code>tfoot</code> , <code>tbody</code> , <code>tr</code> , <code>td</code> , <code>th</code> .

5.2. The `iframe` editor dialog box

By default, an `iframe` element is rendered by embedding a Web browser in the styled view.

Figure 2. An embedded web browser displaying a sample YouTube™ video

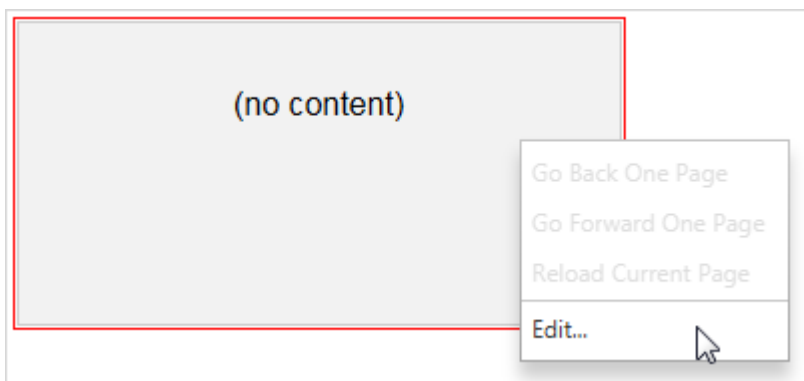


Note

This feature, which is based on JavaFX®, will not work if the Java™ runtime used to run XFE is older than version 1.8.0_40. Moreover, this feature may be explicitly disabled by unchecking Options → Preferences, General|Feature, "In the styled view, render audio and video elements by embedding a media player" in *XMLmind XML Editor - Online Help*.

A newly embedded Web browser is not configured yet (e.g. does not point to any HTML content) and just looks like an empty gray rectangle.

Figure 3. The right-click popup menu of the embedded Web browser

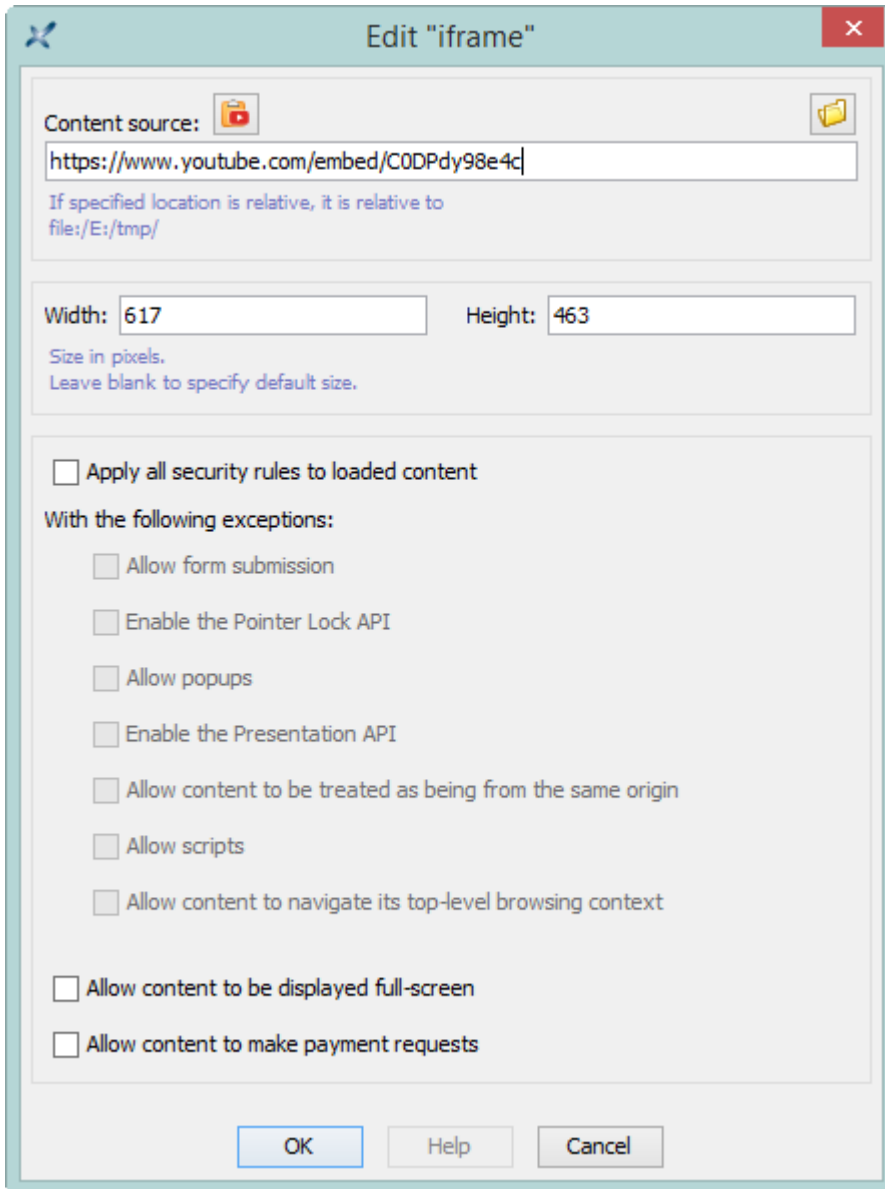


In order to configure a media player, you may either:

- double-click inside it;
- OR right-click in it and select the Edit menu item;
- OR drag and drop an HTML file onto it.

The three above actions all display the `iframe` editor dialog box.

Figure 4. the `iframe` editor dialog box



An `iframe` element is configured as follows:

1. Specify the location of the HTML page to be displayed by the embedded Web browser. This location may be an absolute or relative URL. A relative URL is relative to the URL of the document being edited.

This field corresponds to the `src` attribute of the `iframe` element.

2. Optionally, give a width and/or a height to the `iframe`. These dimensions are specified in pixels. The default size is 300x200. When a single dimension is specified, the other dimension is automatically computed using a 2:1 width/height ratio.


These fields corresponds to the `width` and `height` attributes of the `iframe` element.

3. Optionally, specify security options for the HTML content of the `iframe` element by checking one or more security option checkboxes.

Note that settings such security options have no effect on the Web browser embedded in the styled view of XXE. These checkboxes correspond to values of the `sandbox` attribute of the `iframe` element. For example, checking "Apply all security rules" alone adds attribute `sandbox=""` to the `iframe` element. Other example, checking "Apply all security rules" then "Allow popups" and "Allow scripts" adds attribute `sandbox="allow-popups allow-scripts"` to the `iframe` element.

Tip

The simplest and quickest way to fill the fields of the above dialog box is:

- a. Using your Web browser, visit the YouTube page containing your video.
- b. Right-click the video and select entry "Copy embed code" from the contextual menu.
- c. Click the  button.

This button parses the content of the clipboard (which expected to contain something like `<iframe...src=..."`) and then uses the results to initialize some of the fields of the above dialog box.

6. Custom bindings

Keystroke	Action
Alt+Shift+Up	Same as menu item Move Up [5].
Alt+Shift+Down	Same as menu item Move Down [5].
Alt+Shift+Left	Same as toolbar button Decrease nesting level [7].
Alt+Shift+Right	Same as toolbar button Increase nesting level [7].
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.
Ctrl+Enter	Add same block after the paragraph, list item or a few other kinds of block which is the ancestor of selected node.
Ctrl+Shift+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 <code>br</code> at caret and moves caret after inserted <code>br</code> .
Application Event	Action
Drop an object.	<ul style="list-style-type: none"> • On a <code>a</code> element having an <code>href</code> attribute, change the value of this attribute to the dropped string. • On an <code>img</code> element, change the value of the <code>src</code> attribute to the dropped string.

Application Event	Action
	<ul style="list-style-type: none"> Elsewhere <ul style="list-style-type: none"> If the object being dropped represents an URL or an absolute filename, open the corresponding document in XMLmind XML Editor. Otherwise, displays a popup menu allowing to paste the dropped text or XML before, into or after the drop location.
Drag one of the “handles” displayed around an image. (The “handles” are displayed after clicking on the image.)	<p>Resize the image, but always preserve its aspect ratio.</p> <p>Pressing Ctrl (Cmd on the Mac) while dragging the handle allows to distort the image.</p>
Drag a separator found between two table columns.	<p>Resize the table column. More precisely, this gives an appropriate percent width (e.g. <code><colgroup width="35%"></code>) to <i>all</i> table columns.</p> <p>Warning</p> <p>If the document containing the table is an XHTML 5 document, then the <code>colgroup</code> element has no <code>width</code> attribute. In such case, the <code>resize table column</code> action has to use the <code>style</code> attribute (e.g. <code><colgroup style="width: 35%"></code>). In practice, this means that unless you select the "Emulate Web Browser" CSS stylesheet [3], you'll not see the table columns correctly resized.</p>

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

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

Parameter	Value	Default Value	Description
<code>apply-css-styles</code>	'no' 'yes'	'yes'	<p>Specifies whether CSS styles specified in XHTML <code>style</code> attributes, <code>style</code> and <code>link</code> elements also apply to the XSL-FO file.</p> <p>Depending on the context, the following CSS properties are converted to their equivalent XSL-FO attributes. The corresponding shorthand CSS properties are supported too. <i>Any other CSS property is ignored.</i> Generated content (<code>:before</code>, <code>:after</code>) is ignored too.</p> <ul style="list-style-type: none"> • <code>margin-top</code>, <code>margin-right</code>, <code>margin-bottom</code>, <code>margin-left</code>. • <code>padding-top</code>, <code>padding-right</code>, <code>padding-bottom</code>, <code>padding-left</code>. • <code>border-top-style</code>, <code>border-right-style</code>, <code>border-bottom-style</code>, <code>border-left-style</code>. • <code>border-top-width</code>, <code>border-right-width</code>, <code>border-bottom-width</code>, <code>border-left-width</code>. • <code>border-top-color</code>, <code>border-right-color</code>, <code>border-bottom-color</code>, <code>border-left-color</code>. • <code>background-color</code>, <code>background-image</code>, <code>background-repeat</code>, <code>background-position</code>. • <code>color</code>. • <code>font-family</code>, <code>font-style</code>, <code>font-weight</code>, <code>font-size</code>. • <code>text-decoration</code>. • <code>text-align</code>. • <code>text-indent</code>. • <code>vertical-align</code>. • <code>line-height</code>. • <code>list-style-type</code>, <code>list-style-position</code>, <code>list-style-image</code>. • <code>width</code>, <code>height</code>. • <code>caption-side</code>. • <code>border-spacing</code>. <p><i>Note that styles specified this way supersede all the other ways to specify the presentation in the output file, that is, parameters like <code>base-font-size</code> [15] or the presentation attributes (<code>xsl:attribute-set</code>) specified in the XSLT stylesheets that generate the XSL-FO file.</i></p> <p>Important</p> <p>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.</p>
<code>base-font-size</code>	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.
<code>external-href-after</code>	String	']'	Appended after the external URL referenced by an <code>a</code> element. Ignored unless <code>show-external-links='yes'</code> [17].

Parameter	Value	Default Value	Description
external-href-before	String	' ['	Separates the text of an a element from its referenced external URL. Ignored unless show-external-links='yes' [17].
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” [18].
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” [18].
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” [18].
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” [18].
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” [18].
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” [18].
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” [18].
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” [18].
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” [18].
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” [18].
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” [18].
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” [18].
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” [18].

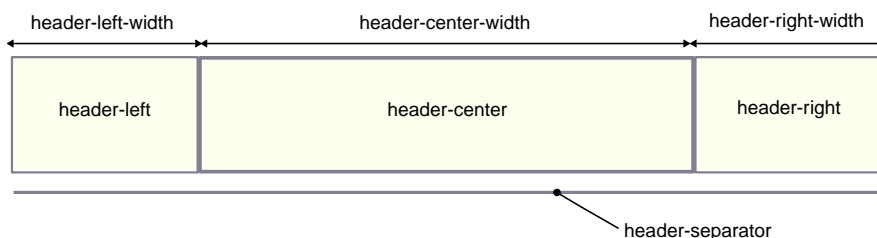
Parameter	Value	Default Value	Description
header-separator	'no' 'yes'	'yes'	Specifies whether an horizontal rule should be drawn below the page header. See the section called “Specifying a header or a footer” [18].
hyphenate	'no' 'yes'	'no'	Specifies whether words may be hyphenated.
justified	'no' 'yes'	'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' [18].
page-ref-before	String	' ['	Separates the text of an a element from the page number it points to. Ignored unless show-xref-page='yes' [18].
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 [20] and page-height [20] parameters.
set-outline-level	'no' 'yes'	'yes'	When XMLmind XSL-FO converter is used to convert the intermediate XSL-FO file to RTF, .odt, .docx, etc, and the value of this parameter is 'yes', consider the h1, h2, ... h6 elements as being headings participating in the outline of the MS-Word or OpenOffice/LibreOffice document. The outline level is directly taken from the name of the heading element: level=1 for h1, level=2 for h2, etc. Note that h1, h2, ... h6 elements contained inside a blockquote, details, dialog, fieldset, figure or td element (that is, a secondary “sectioning root”) never participate in the outline of the MS-Word or OpenOffice/LibreOffice document.
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/].

Parameter	Value	Default Value	Description
show-map-links	'no' 'yes'	'yes'	Specifies whether a numbered list should be generated for a XHTML <code>map</code> element, with one list item per <code>area</code> element. A list item contains the link specified by the <code>area</code> element. No list items are generated for “dead areas” (<code>area</code> elements specifying no link at all).
show-xref-page	'no' 'yes'	'no'	Specifies whether the page number corresponding to the internal link target referenced by an <code>a</code> element should be displayed right after the text contained by this element. Example: <code>show-xref-page='yes'</code> causes <code>Introduction</code> to be rendered as follows: Introduction [3].
two-sided	'no' 'yes'	'no'	Specifies whether the document should be printed double sided.
ul-li-bullets	One or more bullet characters separated by spaces	' • ; – ;'	Specify which bullet character to use for an <code>ul/li</code> element. Additional characters are used for nested <code>li</code> elements. For example, if <code>ul-li-bullets="* - +"</code> , "*" will be used for <code>ul/li</code> elements, "-" will be used for <code>ul/li</code> elements contained in a <code>ul/li</code> element and "+" will be used for <code>ul/li</code> elements nested in two <code>ul/li</code> elements.
xfc-render-as-table	A string containing zero or more element names separated by whitespace. Supported element names are: <code>aside</code> , <code>blockquote</code> , <code>footer</code> , <code>header</code> , <code>nav</code> .	'aside blockquote'	Specifies whether XMLmind XSL-FO Converter should render the <code>fo:blocks</code> representing specified elements as <code>fo:tables</code> . This parameter enables a workaround for a limitation of XMLmind XSL-FO Converter: a <code>fo:block</code> having a border and/or background color and containing several other blocks, lists or tables is very poorly rendered in RTF, WML, DOCX and ODT.

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` [16], `header-center-width` [16], `header-right-width` [16] parameters for the header and the `footer-left-width` [16], `footer-center-width` [16], `footer-right-width` [16] 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` [16], `header-center` [16], `header-right` [16] parameters for the header and the `footer-left` [16], `footer-center` [16], `footer-right` [16] 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}}`

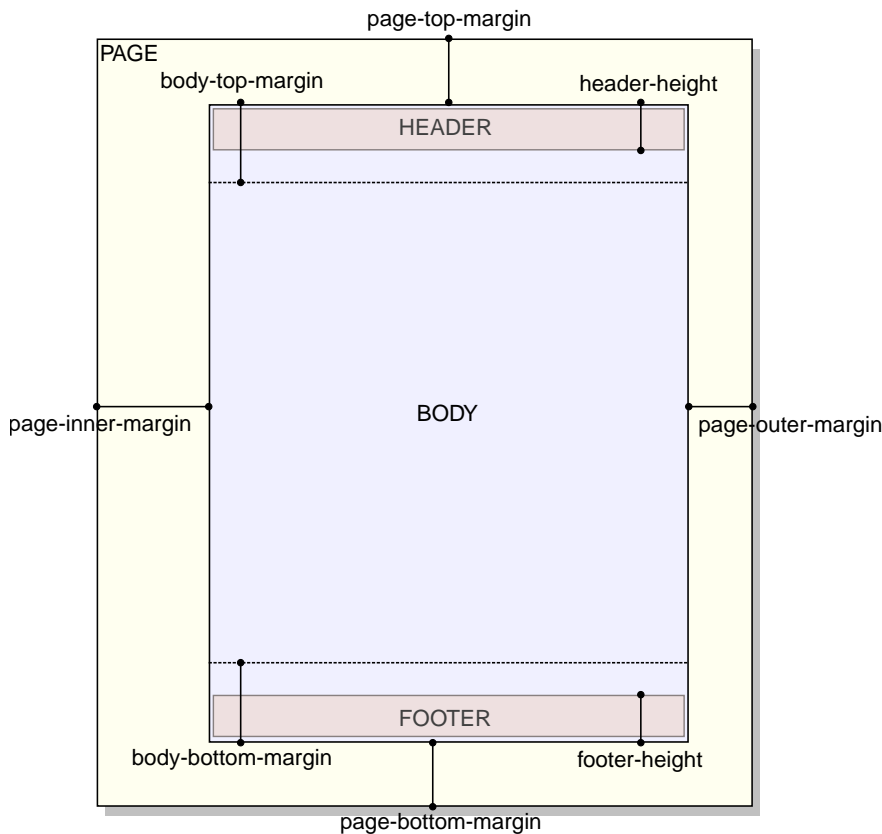
A line break.

`{{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` [20], `body-top-margin` [20], `footer-height` [20], `body-bottom-margin` [20].

Figure B.2. Page areas



Advanced parameters

Parameter	Value	Default Value	Description
img-src-path	URI ending with '/'	' '	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. This low-level alternative to <code>resolve-img-src='yes'</code> [21] also allows to generate an XSL-FO file where all the references to graphic files are absolute URIs.
resolve-a-href	'no' 'yes'	'no'	In the XSL-FO file, convert relative URIs contained in the <code>href</code> attribute of <code>a</code> elements to absolute URIs. This is done by resolving the relative URI against the base of the <code>a</code> element.
resolve-img-src	'no' 'yes'	'yes'	In the XSL-FO file, convert relative URIs contained in the <code>src</code> attribute of <code>img</code> elements to absolute URIs. This is done by resolving the relative URI against the base of the <code>img</code> element.
root-id	ID found in the source XHTML document	' '	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.

Parameter	Value	Default Value	Description
screen-resolution	Number	96.0	Screen resolution in DPI. Used to convert px to pt.

System parameters

Note

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, in some configuration files or in the source code of the XSLT stylesheets.

Parameter	Value	Default Value	Description
foProcessor	String. Examples: 'FOP', 'XEP', 'XFC', 'AHF'.	If <code>outputFormat</code> [22] has been specified and if system property <code>XSL_FO_PROCESSORS</code> 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.