
XMLmind XML Editor - Online Help

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

December 2, 2005

Abstract

This online help is essentially the reference manual of XMLmind XML Editor (XXE for short) menus, tool bars and dialog boxes.

Table of Contents

1. Menus	2
1.1. File menu	2
1.2. Select menu	4
1.2.1. The selection in XXE	4
1.2.1.1. Text selection	4
1.2.1.2. Node selection	5
1.2.1.3. Implicit element selection	6
1.2.2. Menu items	6
1.3. Edit menu	7
1.3.1. Text menu	12
1.3.2. Comment menu	12
1.3.3. Processing instruction menu	13
1.3.4. Document Reference menu	13
1.4. Search menu	14
1.5. View menu	14
1.6. Tools menu	15
1.6.1. Spreadsheet menu	15
1.6.2. "Record Macro" menu	16
1.7. XML menu	17
1.8. Options menu	17
1.9. Window menu	18
1.10. Help menu	18
2. Main tool bar	19
3. Node path/node selection bar	20
3.1. Document reference tool bar	20
3.2. Node Path bar	20
3.2.1. Non-editable nodes	21
3.2.2. Elements which are in non-validating mode	21
3.3. Select tool bar	22
4. Tools	22
4.1. Edit tool	22
4.2. Attributes tool	24
4.2.1. Using the attribute table	25
4.2.2. Using the attribute form	26
4.3. Search tool	27
4.4. Spell tool	29
4.5. Character tool	30
4.5.1. The "Favorites" palette	30
4.6. Validity tool	31
5. Status bar	31
5.1. Validity state	31
5.2. Status messages	32

5.3. Clipboard utilities	32
5.3.1. Clipboard content	32
6. Dialog boxes	33
6.1. The "Options" dialog box	33
6.1.1. Open options	33
6.1.1.1. WebDAV options	35
6.1.1.2. Schema cache options	36
6.1.2. Save options	37
6.1.3. Print options	40
6.1.4. Edit options	41
6.1.5. View options	44
6.1.6. Tools options	45
6.1.6.1. Spell options	45
6.1.6.2. Spreadsheet options	46
6.1.7. Window options	47
6.1.8. General options	47
6.2. The "Declare Namespace" dialog box	48
6.3. The URL chooser dialog box	49
6.4. The "Resources" dialog box	52
6.5. The formula editor	53
6.6. The "Find Element" dialog box	54
6.6.1. The "Simple" tab	54
6.6.2. The "Advanced" tab	55
A. Command line options	55

1. Menus

1.1. File menu

New

Creates a new document which is a copy of a named template.

This dialog box displayed by this command contains a list of named document templates (example: Article). A document template is listed below the name of the XML application (example: DocBook) to which it belongs.

Open Copy

Creates a new document using an existing document as a template. Displays standard file chooser dialog box or advanced URL chooser dialog box depending on value of option "Use URL chooser rather than file chooser".

Unlike New which uses a named template declared in a XXE configuration file, Open Copy uses any existing document as a template.

Open as Template

Creates a new document using an existing document as a template. Displays standard file chooser dialog box or advanced URL chooser dialog box depending on value of option "Use URL chooser rather than file chooser".

Unlike New which uses a named template declared in a XXE configuration file, Open as Template uses an existing document to dynamically create a template. After loading a copy of the existing document, all the content of the copy is discarded to just keep a *valid skeleton* of the root element.

Open

Opens an existing document. Displays standard file chooser dialog box or advanced URL chooser dialog box depending on value of option "Use URL chooser rather than file chooser".

Tip

It is also possible to open a document by dropping it anywhere inside XXE main window (except above an image in a document view).¹

Note

XXE cannot load non-well formed XML documents but it can load invalid documents. See Validity State.

Note

XXE can load a non-XML (though structured) document if a plug-in has been written to support this non-XML format. See Help|Plug-ins.

Save

Saves current document, but only if it has been modified.

Behaves like Save As if current document is a newly created document or if current document has not the write permission for the user.

Tip

At least for XXE, you do not have to save XML documents with a `.xml` extension. You can use more meaningful extensions.

Save As

Saves current document to a different location. Displays standard file chooser dialog box or advanced URL chooser dialog box depending on value of option "Use URL chooser rather than file chooser".

If the document to be saved has resources which are logically part of this document (typically images), a specialized dialog box listing all these resources will be displayed. The user will have to specify what to do with each resource: copy it to the new location and/or update the reference to the resource in the newly saved document.

Note

These resources are specified for each document type in the corresponding XXE configuration file using a `documentResources` configuration element. See *XMLmind XML Editor - Configuration and Deployment*.

Save Copy

Similar to Save As except that it saves a copy of current document. Displays standard file chooser dialog box or advanced URL chooser dialog box depending on value of options "Use URL chooser rather than file chooser" and "Always use the URL chooser to save a copy".

Save All

Saves all modified documents.

Page Setup

Selects the page size and margins used for printing. (Displays standard Page Setup dialog box.)

Print

Prints current view of current document. (Displays standard Print dialog box.)

¹Linux users: Java™ 1.4 only supports the Motif drag and drop protocol. Therefore drag and drop probably does not work between XXE and your favorite Gnome or KDE application.

Tip

It is possible to print a subset of current document by explicitly selecting the element to be printed. See Select menu.

Quit

Quit XXE.

Below the above menu items, a menu item is added for the last nine documents that have been loaded into XXE.

Selecting such menu item:

- Loads the document into XXE if it is not currently loaded.
- Reloads the document into XXE, closing all existing views, if it is currently loaded. If the document has been modified, the user will have to confirm that changes are to be discarded.

1.2. Select menu

1.2.1. The selection in XXE

There are three types of selection in XXE:

- text selection (also called character selection),
- node selection (a node is either a named element or an anonymous text node or an anonymous comment node or an anonymous processing instruction node),
- implicit element selection.

1.2.1.1. Text selection

The text selection is the type of selection supported by all text editors and word processors.

It is, of course, possible to select text across document nodes.

Selected text is displayed as characters drawn over a pink background.

The text selection can be seen as a way to specify a range of characters and descendant nodes contained *in a common ancestor node*. The copy, cut, delete, paste, convert commands will copy, delete or replace the specified characters and descendant nodes.

- Example 1: "`<p>This is our new logo.</p>`". Selected text starts at "This" and ends after "new" and is to be deleted.

This delete operation gives "`<p> logo.</p>`".

- Example 2: "`<p>This is great, really!</p>`". Selected text starts at "This" and ends after "great" and is to be converted to ``.

This convert operation gives "`<p>This is great, really!</p>`".

How to select text in XXE:

Using the mouse	Using the keyboard
<ul style="list-style-type: none"> • Click at the beginning of the text selection and drag the mouse to the end of selection. 	<ul style="list-style-type: none"> • Shift-Left or Shift-Right extends the text selection by one character. • Ctrl-Shift-Left or Ctrl-Shift-Right extends the text selection by one word.

Using the mouse	Using the keyboard
	<ul style="list-style-type: none"> • Ctrl-Shift-Home extends the text selection to the beginning of the line. • Ctrl-Shift-End extends the text selection to the end of the line. • Shift-Down or Shift-Up extends the text selection by one line.

1.2.1.2. Node selection

The node selection is unique to XML editors.

In XXE, you can even select several nodes at the same time if these nodes are consecutive children of the same parent (contiguous range of child nodes).

Selected nodes are displayed with a thin red border around them.

How to explicitly select a *single node* in XXE:

Using the mouse	Using the keyboard
<ul style="list-style-type: none"> • Click on the node. If some text is near the place where you have clicked, it will "attract" the caret and the node will not be selected. In such case, you are forced to Ctrl-click instead of simply clicking. • Or Ctrl-click on the node. Ctrl-click several times (without moving the mouse) if needed to. Each Ctrl-click will select the parent of currently selected node. Do not Ctrl-click several times too fast otherwise the editor will think you are double-clicking or triple-clicking and therefore, selecting elements that way would not work. • Or click on generated content such as a list bullet or a section number. • Or Ctrl-click in the blank space found at the right of the text of a paragraph. • Or click on the name of the element you want to select in the Node Path bar. 	<ul style="list-style-type: none"> • Click inside the node to move the caret there. Type Ctrl-Up until you reach the node you want to select. Keep an eye on the Node Path bar while doing this. • Or use the Select menu or the Select tool bar or any of the corresponding keyboard shortcuts Ctrl-Up, Ctrl-Down, Shift-Ctrl-Up or Shift-Ctrl-Down.

Procedure for selecting a *node range* using the *mouse*:

1. Select first node using any of the methods described above.

Make sure that you have selected the right node by looking at the Node Path bar, otherwise extending the node selection will not work.

2. Shift-Ctrl-click on the last node of the selection.

If there is no ambiguity, you can even Shift-Ctrl-click *anywhere* past the last node of the selection.

Procedure for selecting a *node range* using the *keyboard*:

- **Esc-Down** selects all child nodes of explicitly or implicitly selected element.
- OR
 1. Select first node using any of the methods described above.

Make sure that you have selected the right node by looking at the Node Path bar, otherwise extending the node selection will not work.
 2. Adjust selected node range: **Esc-Right** extends node selection to following sibling and **Esc-Left** extends node selection to preceding sibling.

Note **Esc-Right** (and **Esc-Left**) will first select element containing caret if there is no explicit node selection, therefore typing **Esc-Right** several times is often the quickest way to select a node range.

1.2.1.3. Implicit element selection

The implicitly selected element is simply the element containing the *caret* (also called the insertion cursor).

Being implicitly selected, this element is not drawn a red border around it but you know it because it is the element which is displayed in the Node Path bar.

All editing commands except the most generic form of Split and Join can be applied to the *implicitly selected element*.

Important

Almost all editing commands do not require you to explicitly select the element you want to act upon.
This makes XXE at the same time efficient and easy to use.

1.2.2. Menu items

Tip

Clicking anywhere in the document view or any caret movement has the side effect to cancel the selection.
It is also possible to explicitly do so by typing **Esc-Esc**.

Menu items:

Select Parent

Selects parent of currently selected node.

If there is no currently selected node, selects the text, comment or processing instruction node containing the caret.

Select Child

Selects previously selected child of currently selected node.

If there is no such child, selects first child node of currently selected element.

If currently selected node is a text, comment or processing instruction node, cancels the selection.

Select Preceding Sibling

Selects preceding sibling of currently selected node, if any.

Select Following Sibling

Selects following sibling of currently selected node, if any.

Extend Selection to Preceding Sibling

Adds preceding sibling, if any, of currently selected node to the node selection.

If there is no currently selected node, selects the element containing the caret.

Keyboard shortcut (not displayed in menu): **Esc-Left**.

Extend Selection to Following Sibling

Adds following sibling, if any, of currently selected node to the node selection.

If there is no currently selected node, selects the element containing the caret.

Keyboard shortcut (not displayed in menu): **Esc-Right**.

Select All Children

Selects all children of currently selected element.

Keyboard shortcut (not displayed in menu): **Esc-Down**.

Find Element

Displays the Find Element dialog box. This dialog box allows to select nodes specified using an XPath expression. This dialog box has a Simple tab which allows to perform most common search tasks without having to learn XPath. Arbitrarily complex XPath expressions are specified using the Advanced tab.

Redraw

Rebuilds the view of currently selected node.

If there is no currently selected node, rebuilds the view of the whole document.

This command is actually a workaround for a minor deficiency of XXE. If a style depends on the position of a element (example: "**listitem > *:first-child {margin-top:0;}**") relatively to its siblings and if the position of this element changes (example: its preceding sibling has been deleted), XXE is not currently able to automatically detect this situation and therefore update the style of the element accordingly. The user has to do that manually by using the Redraw command.

This generally happens in the following cases:

- For the first child paragraph of a list item: its margins are incorrect.
- For the last cell of a row group or column group: its borders are incorrect.

1.3. Edit menu

Tip

A simplified Edit menu will popup if the right mouse button is clicked anywhere in a document view. Some of the menu items of this popup Edit menu have more accurate titles than those of the menu bar Edit menu (example: "Undo Convert" instead of simply "Undo").

Undo

Undo last command.

Redo

Redo last undone command.

Repeat

Repeats last repeatable command.

Commands requiring the user to specify an argument (e.g. Replace, Insert Before, Insert, Insert After, Convert, Convert [wrap], Change Processing Instruction Target, etc) are repeatable.

Command History

Displays a dialog box listing last ten repeatable commands from newest to oldest.

Cut

Cuts

- text selection
- OR explicitly selected node or node range
- OR implicitly selected element

to system clipboard.

Tip

It is possible to cut and paste nodes between two instances of XXE (of course if the DTD or schema allows it).

Copy

Copies

- text selection
- OR explicitly selected node or node range
- OR implicitly selected element

to system clipboard.

Tip

Selected characters are automatically copied as system selection on platforms supporting system selection (X-Window) and automatically copied to an internal clipboard on other platforms.

Paste Before

Pastes the content of system clipboard before

- explicitly selected node or node range
- OR implicitly selected element.

The system clipboard may contain one or several nodes or just plain text. The content of system clipboard, is parsed as XML if it begins with "<?xml" otherwise it is considered to be plain text.

Paste

Pastes the content of system clipboard replacing

- text selection
- OR explicitly selected node or node range,

OR if there is no explicit selection, pastes the content of system clipboard into

- element containing caret, at caret position.

The system clipboard may contain one or several nodes or just plain text. The content of system clipboard, is parsed as XML if it begins with "<?xml" otherwise it is considered to be plain text.

Tip

Clicking with mouse button #2 (middle button or mouse wheel) can be used to paste the content of system selection on platforms supporting system selection and can be used to paste the content of an internal clipboard on other platforms (if allowed by grammar constraining the document, of course).

By default, this very handy feature is not enabled. You need to enable it using the Options dialog box.

Paste After

Pastes the content of system clipboard after

- explicitly selected node or node range
- OR implicitly selected element.

The system clipboard may contain one or several nodes or just plain text. The content of system clipboard, is parsed as XML if it begins with "<?xml" otherwise it is considered to be plain text.

Delete

Deletes

- text selection
- OR explicitly selected node or node range
- OR implicitly selected element.

Force Deletion

Like Delete except that deletion will be performed even if the grammar constraining the document forbids to do so.

Example 1. Example of use:

The content model of element <a> is child element or a sequence of child element <c> followed by child element <d>.

A new <a> is by default created with the simplest possible content model, that is . Then how to replace by the sequence <c><d>? Deleting is forbidden because it would give us an invalid <a>.

The answer is:

1. Force the deletion of using the command described here. This makes <a> temporarily invalid but also relaxes the constraints on it.
2. Insert a <c>.
3. Insert a <d>. Element <a> is now valid.

Replace

Displays the Edit tool which can be used to specify an element replacing

- explicitly selected node or node range
- OR implicitly selected element.

A dialog box can be displayed rather than the Edit tool if option "Use dialog box rather than Edit tab" is used.

Insert Before

Displays the Edit tool which can be used to specify an element inserted before

- explicitly selected node or node range
- OR implicitly selected element.

A dialog box can be displayed rather than the Edit tool if option "Use dialog box rather than Edit tab" is used.

Insert

Displays the Edit tool which can be used to specify an element inserted into

- explicitly selected *empty* element
- OR element containing caret, at caret position.

A dialog box can be displayed rather than the Edit tool if option "Use dialog box rather than Edit tab" is used.

Insert After

Displays the Edit tool which can be used to specify an element inserted after

- explicitly selected node or node range
- OR implicitly selected element.

A dialog box can be displayed rather than the Edit tool if option "Use dialog box rather than Edit tab" is used.

Convert

Displays the Edit tool which can be used to specify an element replacing

- text selection
- OR explicitly selected node or node range
- OR implicitly selected element.

A dialog box can be displayed rather than the Edit tool if option "Use dialog box rather than Edit tab" is used.

Unlike Replace which creates an *empty* new element, Convert transfers the content of the selection to the new element which is the result of the conversion.

More precisely, in the case of the node selection:

- When a single element is selected, all its children (but not its attributes) are transferred to the result of the conversion.

Example:

```
"<simpara>The <emphasis>little</emphasis> lamb.</simpara>"
```

converted to <para> gives

```
"<para>The <emphasis>little</emphasis> lamb.</para>".
```

- When several nodes or a single non-element node are selected, all these nodes are given a new parent element which is the result of the conversion.

Example:

```
"<simpara>Once upon a time,</simpara>"
```

plus

```
"<simpara>the <emphasis>little</emphasis> girl.</simpara>"
```

can be converted to `<blockquote>` and that gives us

```
"<blockquote><simpara>Once upon a time,</simpara><simpara>the <emphasis>little</emphas-  
is> girl.</simpara></blockquote>"
```

See also [Wrap](#) a variant of [Convert](#).

Convert [wrap]

This command is a variant of [Convert](#). The unique difference between [Wrap](#) and [Convert](#) is that, with [Wrap](#), when a single element is selected, the selected element is given a new parent element.

Example, with [Wrap](#) (and not with [Convert](#)), it is possible to give a `<blockquote>` parent to the following `<simpara>`, when this `<simpara>` is implicitly or explicitly selected:

```
"<simpara>The <emphasis>little</emphasis> lamb.</simpara>"
```

That is, selecting `<blockquote>` using the [Edit](#) tool gives:

```
"<blockquote><simpara>The <emphasis>little</emphasis> lamb.</simpara></blockquote>"
```

Split

Splits explicitly selected element in two parts, the split point being specified by caret position.

Unlike almost all other commands, this command requires the element to be explicitly selected.

Keyboard shortcut (not displayed in menu): **Esc-Enter**.

Tip

A less generic form of the [Split](#) command is *often* bound to key [Enter](#) (for example, this is the case for [XHTML](#) and [DocBook](#)).

Typing [Enter](#) inside a paragraph (that is, `<p>` for [XHTML](#) and `<para>` or `<simpara>` for [DocBook](#)) will split this element in two parts.

Example: `<simpara>` is explicitly selected and the caret is in the middle of word "little". Splitting

```
"<simpara>the <emphasis>little</emphasis> girl.</simpara>"
```

gives us

```
"<simpara>the <emphasis>lit</emphasis></simpara><simpara><emphasis>tle</emphasis>  
girl.</simpara>"
```

Tip

Therefore, typing [Enter](#) at the end of a paragraph will create an empty new paragraph after it.

Typing [Enter](#) at the beginning of a paragraph will create an empty new paragraph before it.

Join

Joins explicitly selected element to its preceding sibling, an element of same type. This gives a single element containing the child nodes of the two joined elements.

This command is the inverse command of [Split](#).

Unlike almost all other commands, this command requires the element to be explicitly selected.

Keyboard shortcut (not displayed in menu): **Esc-BackSpace**. It is also possible to type **Esc-Del** to join explicitly selected element to its *following* sibling, an element of same type.

Tip

A less generic form of the Join command is *often* bound (for example, this is the case for XHTML and DocBook) to

- key Backspace when the caret is at the beginning of a paragraph
- and to key Delete when the caret is at the end of a paragraph.

Typing Backspace at the beginning of a paragraph joins this element to the preceding paragraph.

Typing Delete at the end of a paragraph joins this element to the following paragraph.

1.3.1. Text menu

Insert Text Before

Inserts text node before

- explicitly selected node
- OR implicitly selected element.

Insert Text

Inserts text node into

- explicitly selected *empty* element
- OR element containing caret, at caret position.

Insert Text After

Inserts text node after

- explicitly selected node
- OR implicitly selected element.

1.3.2. Comment menu

Insert Comment Before

Inserts comment node before

- explicitly selected node
- OR implicitly selected element.

Insert Comment

Inserts comment node into

- explicitly selected *empty* element
- OR element containing caret, at caret position.

Insert Comment After

Inserts comment node after

- explicitly selected node
- OR implicitly selected element.

1.3.3. Processing instruction menu

Insert Processing Instruction Before

Inserts processing instruction node (with a target called "target") before

- explicitly selected node
- OR implicitly selected element.

Insert Processing Instruction

Inserts processing instruction node (with a target called "target") into

- explicitly selected *empty* element
- OR element containing caret, at caret position.

Insert Processing Instruction After

Inserts processing instruction node (with a target called "target") after

- explicitly selected node
- OR implicitly selected element.

Change Processing Instruction Target

Displays a dialog box that can be used to change the target of

- explicitly selected processing instruction node
- OR implicitly selected processing instruction node (that is the processing instruction node containing the caret).

1.3.4. Document Reference menu

Copy as Reference

Copies to the clipboard a *reference* to the explicitly or implicitly selected element.

This reference can be later pasted in another document, using the normal paste command (**Ctrl-V**), in places where the grammar constraining the target document allows to do so.

It is possible to "copy as reference" an element reference already pasted in a document. This is handy because it sometimes spares the effort of switching from the view of the referencing document to the view of the referenced document.

It is not possible to "copy as reference" a descendant element found deep inside an element reference already pasted in a document.

Note

- The pasted element reference cannot be edited in place. It is displayed with a light gray background to clearly indicate this. Use command Edit|Document Reference|Edit Referenced Document (or tool bar button ) to open a new window allowing to edit the referenced document.
- Specifying how a reference to an element contained in an other document is to be saved can be specified (to a certain extent) using the Options dialog, Edit section.

Edit Referencing Document

If current document is referenced by another document already loaded in XXE and displayed by another window, this command brings all the windows of this other document to front. If there is no such referencing document, this command is silently disabled.

Example 2. Book with referenced chapters example

For example, if a book document `book.xml` references chapters `chap1.xml`, `chap2.xml`, `chap3.xml`, etc, created in separate documents.

Clicking anywhere inside first chapter displayed by `book.xml` window and using command `Edit|Document Reference|Edit Referenced Document` (or tool bar button ) will bring all windows containing `chap1.xml` to front. (If needed, `chap1.xml` is opened.)

Now being inside `chap1.xml` window, using command `Edit|Document Reference|Edit Referencing Document` (or tool bar button ) will bring all windows containing `book.xml` to front.

As of XXE V2.6, the views of referencing documents are not automatically refreshed after modifications made to referenced documents. If you need to refresh the view of a referencing document, you'll have manually reload this document in XXE. A quick way to do so is to drag in the document view the bookmark-like icon found in the node path bar.

Edit Referenced Document

If explicitly or implicitly selected element or any of its ancestors is a reference to an element contained in another document, this command brings all the windows of this other document to front. If the referenced document is not yet opened in XXE, this command will open it.

See example above.

1.4. Search menu

Search

Displays Search tool configured for a search session from caret to end of current document.

Search Backwards

Displays Search tool configured for a search session from caret to beginning of current document.

Replace

Displays Search tool configured for a search/replace session from caret to end of current document.

Replace Backwards

Displays Search tool configured for a search/replace session from caret to beginning of current document.

Find Next

Search last searched string from caret to end of current document. Search tool is not displayed because it is not needed.

Find Previous

Search last searched string from caret to beginning of current document. Search tool is not displayed because it is not needed.

1.5. View menu

Add

Opens a dialog which allows to add a new view to current document tab. A document tab can contain up to 5 views: default central view, but also top, right, bottom, left views. A view is specified by selecting a CSS style sheet among the available ones or, on the contrary, by selecting no style sheet at all, which implies to use a tree view.

Close

Closes active view. Central view, which is supposed to be the main view, cannot be closed.

The active view is the view having the keyboard focus: the caret blinks in this view, and not in the other views of the document. To make a view the active one, simply click anywhere in it.

Below the above menu item, a menu item is added for each CSS style sheet available for current document. Selecting the name of a style sheet causes the document view to use this style sheet. If a document view already uses the selected style sheet, the style sheet is reloaded from its file (which is very handy when developing a new CSS style sheet).

1.6. Tools menu

Declare Namespace

Displays Declare Namespace dialog box.

This command is disabled if current document uses a DTD as its grammar. See Namespace support in XXE.

Edit Attribute

Displays Attributes tool.

Check Validity

Displays Validity tool, unless no validity errors are found in current document, in which case an OK message is displayed in the status bar.

This command is disabled if current document is not constrained by a grammar.

Tip

Current document validity is automatically checked each time the document is saved, therefore unless you are fixing an invalid document you don't really need to explicitly use this command.

Check Spelling

Displays Spell tool, unless no spelling errors are found in current document (starting from caret position), in which case an OK message is displayed in the status bar.

Execute Command

For advanced users only. Displays a dialog box which allows to choose a command by name. This is needed when a command is not bound to a keystroke, menu item or tool bar button. Example: command `convertCase` (see XMLmind XML Editor - Commands). This facility is especially useful when recording a macro-command.

1.6.1. Spreadsheet menu

Show Table Labels, Hide Table Labels

Makes tables look like spreadsheets by adding A1-style labels to their columns and to their rows. This is very useful when you want to type cell references such as A1:C3 in a formula.

Insert/Edit Formula

If a formula (`xxe-formula` processing instruction) is explicitly selected, a specialized formula editor is opened to allow the user to modify the formula.

Otherwise, a specialized formula editor is opened to allow the user to insert a new formula at caret position. Inserting a new formula in an empty element works too: simply explicitly select this empty element and use this menu item.

Disable Selected Formulas, Enable Selected Formulas

Disables/enables all the formulas (`xxe-formula` processing instruction) found in the explicit node selection or in the text selection.

Disabling a formula means passivating it. That is, it is no longer used to update the document. In some cases, this is a handy alternative to removing it.

Tip

In the styled view, formulas are represented by a small F icon. Clicking on this icon with the middle button of the mouse allows to switch the state of the formula from enabled  to disabled  and vice-versa.

Remove Selected Formulas

Remove all the formulas (`xxe-formula` processing instruction) found in the explicit node selection or in the text selection.

Clear External Document Cache

Clears the cache of external documents accessed by formulas.

Formulas can access external documents using *XPath escapes* (example: ``document("tutorial/VAT-rates.html#france_vat",.)``). Such access is generally slow and therefore, documents need to be cached the first time they are loaded.

However, the cache is not very smart and will often not be able to detect changes in the external document. This problem will happen if the change happens in a module included by the document or if the document is stored on a remote HTTP or FTP server. In this case, disabling the cache and also disabling the auto-update mode are recommended.

See also Spreadsheet options.

Auto-update Mode

This toggle may be used to switch from auto-update mode to manual update mode.

In manual update mode, only newly inserted formulas are computed. To force a full calculation, the user has to explicitly use the Update command below.

In auto-update mode, a full calculation is automatically performed, if needed to, when the editing context changes. For example: type some text in a paragraph, then click in (or tab to) another paragraph to trigger a spreadsheet calculation.

Note that in both modes, a full calculation is automatically performed, if needed to, before validating or saving the document.

Using manual update mode is recommended if you have a slow computer or if you have inserted a lot of formulas in your document or if your formulas access many external documents.

See also Spreadsheet options.

Update

Forces a full calculation of the spreadsheet.

1.6.2. "Record Macro" menu

This menu allows to record a sequence of commands and to replay the recorded sequence at will.

This facility used in conjunction with Select|Find Element facility or with the Search tool may be seen as an advanced, versatile, yet simple to use, form of search/replace.

Start

Starts recording a sequence of commands.

Stop

Stops recording the sequence of commands.

Cancel

Cancels the recording of a sequence of commands.

View

Displays a dialog box containing recorded macro in XML form. Very handy to paste it in an XXE configuration file (see XMLmind XML Editor - Configuration and Deployment).

Replay

Replays recorded sequence of commands.

Procedure 1. Procedure for recording a sequence of commands

1. Use the Start menu item.
2. Invoke commands as usual: use key bindings, menu items and tool bar buttons.

Typing some text is of course supported. ``Tabbing" from a text node to another is supported too.

The following tools can also be used during a recording: Edit, Attributes, Search, Characters.

If you use the Search tool during a recording, just use the search part; do not use the replace part. And after the searched string is found and selected, click on the Stop button of the Search tool to give keyboard focus back to the document view. This will allow you to continue adding more commands to the recorded sequence.

3. Use the Stop menu item.

Only *editing commands* can be recorded. That is, commands that modify the contents and/or the selection marks of the document being edited. Actions such as File|Open or Style|Both Tree and Styled Views cannot be recorded.

At most 20 commands can be recorded. Typing contiguous characters, no matter how many, counts as a single command.

Attempting to record the following commands will automatically cause macro recording to be canceled:

- any command triggered by a mouse click,
- Undo, Redo, Repeat,
- any command which fails (example: searching a string and this string is not found),
- any command which cannot be executed given current editing context (example: pasting some text to a place where the schema forbids to do so).

Recording interactive command such as Insert After works as expected: it is the command *along with the element interactively chosen by the user* which is recorded, and not the interactive invocation of Edit|Insert After (i.e. which activates the Edit tool or which displays the equivalent dialog box).

Recording command Execute Command is fully supported and works as expected: it is the command executed by Tools|Execute Command which is recorded, and not the invocation of Tools|Execute Command.

1.7. XML menu

This menu is a *placeholder* for menu specific to an XML applications, possibly defined in an XXE configuration file.

For example, such menu is defined for XHTML. When an XHTML document is loaded into XXE, the XML menu is automatically populated with items and its title changes from "XML" to the title declared in the configuration file (example: "XHTML").

1.8. Options menu

Options

Displays the Options dialog box.

Reload All Configurations

Reloads all configuration files found by XXE at startup time. This command is disabled if one or more documents are opened in the editor.

Without this command, testing modifications made to an existing configuration requires you to restart the editor. Note that restarting the editor is still required to make it discover new configurations or to reload support code.

Show Attribute List in Tree View

Toggles the display of the attribute list in the tree view. When this option is turned off, the tree view just displays the number of attributes of an element.

Show Text Characters in Tree View

Toggles the display of text node characters in the tree view. When this option is turned off, the tree view represents a text node as an icon.

Show Comment Characters in Tree View

Toggles the display of comment node characters in the tree view. When this option is turned off, the tree view represents a comment node as an icon.

Show PI Characters in Tree View

Toggles the display of processing instruction node characters in the tree view. When this option is turned off, the tree view represents a processing instruction node as an icon.

Split Windows Vertically

Clicking on the dashed line of a document tab (also called a document window) causes the window area to be split in two parts. This allows to see two documents side by side.

If this option is turned on, the window area is split vertically. If this option is turned off (the default), the window area is split horizontally.

Note that turning this option on and off has an immediate effect on the window area, if this area is already split in two parts.

1.9. Window menu

Close

Closes active document.

Close All

Closes all opened documents.

Show Preceding

Displays preceding (in the order the documents have been opened or created) document tab.

Show Following

Displays following (in the order the documents have been opened or created) document tab.

Below the above menu items, a menu item is added for each document opened in XXE. Selecting the name of a document in this list causes this document to become the active one and thus, causes the corresponding tab to be displayed.

1.10. Help menu

Help

Displays the help browser.

Contextual Help

To get help about a specific GUI component of XXE, execute this command (the cursor changes signaling that you are now in contextual help mode) and then click on that component.

Show Content Model

Opens a window containing an hypertext reference manual listing all elements and attributes specified in the DTD, W3C XML Schema or RELAX NG schema of the document being edited.

This manual, which is organized like "*DocBook: The Definitive Guide*" by Norman Walsh and al., is intended to help content authors understand the DTD or schema of the document being edited.

Mouse and Key Bindings

Displays a dialog box containing the mouse and key *bindings* (that is, the mouse or keyboard user interaction used to trigger a command) that can be used in current document view.

Note

An XXE configuration file, which is a customization of XXE for a specific XML application, can add new commands and/or new mouse and keys bindings to the default set. Therefore the information displayed in this dialog box depends on the type of current document.

Plug-ins

Displays a dialog box containing information about all plug-ins currently loaded into XXE.

About XMLmind XML Editor

Displays the customary About dialog box.

2. Main tool bar



New

See File|New.



Open

See File|Open.



Save

See File|Save.



Save All

See File|Save All.



Use URL chooser rather than file chooser

See option Use URL chooser rather than file chooser.



Undo

See Edit|Undo.



Redo

See Edit|Redo.



Repeat

See Edit|Repeat.



Command History

See Edit|Command History.



Cut

See Edit|Cut.



Copy

See Edit|Copy.

 Paste Before
See Edit|Paste Before.

 Paste
See Edit|Paste.

 Paste After
See Edit|Paste After.

 Delete
See Edit|Delete.

 Split
See Edit|Split.

 Join
See Edit|Join.

3. Node path/node selection bar

3.1. Document reference tool bar

 Edit Referencing Document
See Edit|Document Reference|Edit Referencing Document.

 Edit Referenced Document
See Edit|Document Reference|Edit Referenced Document.

3.2. Node Path bar

The node path bar displays the *path* of

- selected node if a single node has been selected
- OR first selected node if a node range has been selected
- OR node containing the caret otherwise (even if there is a text selection).

The path of a node is

- the name of the element if the node is an element,
- #text for a text node,
- #comment for a comment node,
- #processing-instruction for a processing instruction node,

preceded by the path of its parent element.

The node path bar is a very convenient tool for selecting nodes, duplicating and deleting elements:

User interaction in the node path bar	Command
Click on node name.	Selects this node.
Ctrl-click on element name.	Selects this element and then creates a new element of same type (if the grammar constraining the document allows to do so) after this element.

User interaction in the node path bar	Command
Shift-click on element name.	Selects this element and then creates a new element of same type (if the grammar constraining the document allows to do so) before this element.
Click on element name with right mouse button.	Displays a popup menu with the following items: Insert Same Element After Same as command triggered by Ctrl-click on element name. Insert Same Element Before Same as command triggered by Shift-click on element name. Delete Selects the node clicked upon and then deletes it (if the grammar constraining the document allows to do so). Copy XPath Selects the node clicked upon and then copies its XPath (example: /section[1]/para[1]/text()[1]) to the clipboard.
Drag the icon which looks like a bookmark	Drag and drop in another application, the <i>location</i> of the document being edited in XXE. ^a Do not forget to save the document being edited in XXE before dropping its location in a viewer application.

^aLinux users: Java™ 1.4 only supports the Motif drag and drop protocol. Therefore drag and drop probably does not work between XXE and your favorite Gnome or KDE application.

3.2.1. Non-editable nodes

The path of a non-editable node is displayed using a dimmed color. A non-editable node has been pasted in the document being edited after using command Copy as Reference.



In order to modify such node, one must edit the document containing it in a separate window. A convenient way to do this is to use the buttons which are at the left of the node path bar.

3.2.2. Elements which are in non-validating mode

Normally, XMLmind XML Editor works in validating mode. In such mode, the author can only do the editing operations which are allowed by the DTD or schema. For example, the author cannot remove required attributes (unless he/she uses Force Removal).

When an element is found to be invalid, XMLmind XML Editor automatically switches to a lenient editing mode for this element and all its descendants.

After the invalid element is fixed by the author, XMLmind XML Editor automatically switches back to its normal, validating, mode.

- An element underlined in *orange* means that this element is in *non-validating mode 1*. In this mode, XMLmind XML Editor still suggests the right attributes and child elements to the author. But these are only suggestions: the author may add and remove any attributes and child elements he/she wants, and this, at any place and in any number.



- An element underlined in *red* means that this element is in *non-validating mode 2*. In this mode, XMLmind XML Editor is not able to suggest the right attributes and the right child elements to the author. The author may add and remove any attributes and child elements he/she wants, at any place and in any number.



Note that, in the case of certain RELAX NG schemas, XMLmind XML Editor may automatically switch to a lenient editing mode, *even if the underlined element is perfectly valid*. This behavior is explained in great details in XMLmind XML Editor - Support of RELAX NG Schemas.

3.3. Select tool bar

- ◆ Select Parent
See Select|Select Parent.
- ♥ Select Child
See Select|Select Child.
- ◆ Select Preceding Sibling
See Select|Select Preceding Sibling.
- ◆ Select Following Sibling
See Select|Select Following Sibling.
- 🔍 Find Element
See Select|Find Element.

4. Tools

It is possible to display two tools at the same time. By default, XMLmind XML Editor displays both the Edit tool and the Attributes tool. If you don't like this layout or if your screen resolution is too low to allow this, simply click on the ``tear-off'' dashed line found in the Edit tab. This will move the Edit tool back to the pane containing all the other tools.

More generally, clicking on the ``tear-off'' dashed line found in a tab may be used to split the tabbed pane in two parts and/or move a tab from one part to the other. Of course, the divider used to separate the two parts may be dragged to resize these parts.

4.1. Edit tool

Note

Turn option "Use dialog box rather than Edit tab" on if you prefer to use a dialog box rather than this tool.

This tool works in two steps:

1. Click on one of the following buttons:

- ◆ Replace
(See Edit|Replace for a complete description of the command.)
- ◆ Insert Before
(See Edit|Insert Before for a complete description of the command.)
- ◆ Insert
(See Edit|Insert for a complete description of the command.)

◆ Insert After
(See Edit|Insert After for a complete description of the command.)

◆ Convert
(See Edit|Convert for a complete description of the command.)

◆ Convert [wrap]
(See Edit|Convert [wrap] for a complete description of the command.)

This first action just fills the list below these buttons with all the allowed arguments (element name or text node) for the selected command.

2. Choose an element name from the list or type its name in the text field. It is this second action which actually triggers the command.

Procedure for specifying an argument for one of the editing commands described above:

- An element is specified by clicking on its name in the list. A single click is sufficient.

If the selected operation allows to specify a text node as its argument, the list contains not only element names but also a "(text)" item.

- Alternatively, the element name can be specified using the text field above the list:
 1. Type the element name in the field (or type "(text)" if this item is found in the list).
 2. Press Enter or click on the ✓ OK button.

Tip

The text field supports auto-completion.

This auto-completion feature can be configured using the Options dialog box.

- Type **Esc** or click on the ✗ Cancel button to cancel current command and to clear the form.

Procedure for specifying an element argument when the command is applied to an element of the *unconstrained* type:

1. Type any element name in the text field.

Note

If the namespace of the element name has not been declared, you will have to declare it using Tools|Declare Namespace before being allowed to specify this name.

2. Press Enter or click on the ✓ OK button.

The type of an element is unconstrained if

- the element is contained in a document which is not constrained by a grammar,
- OR the content of the element is invalid according to the grammar used to constrain the document.
- OR the content of an ancestor of the element is invalid according to the grammar used to constrain the document.

In all cases, XXE allows the user to add or remove any child element or text node.

In the second case, fixing the invalid element will automatically make it recover its actual, constrained, type.

Procedure for specifying an element argument that matches an *element wildcard*:

1. Type an element name that matches the wildcard in the text field.

Note

If the namespace of the element name has not been declared, you will have to declare it using Tools|Declare Namespace before being allowed to specify this name.

2. Press Enter or click on the  OK button.

XXE will of course check that the name you have typed conforms to the element wildcard.

Using a W3C XML Schema or a RELAX NG schema (but not a DTD), it is possible to specify an element type which constrains some of its child elements just to belong to a certain set of namespaces and nothing more.

The set of allowed namespaces is called a wildcard.

4.2. Attributes tool

- There are two methods for adding or changing the attributes of the (explicitly or implicitly) selected element:
 1. Using the attribute form (the upper side of the attribute tool). This should be the method of choice for persons who prefer to use the keyboard.
 2. Using the attribute table (the lower side of the attribute tool). This should be the method of choice for persons who prefer to use the mouse.
- The "minus" button of the attribute form can be used to remove an attribute. Removing an attribute directly from the attribute table is possible too: simply right-click on the attribute row and use the popup menu.
- Adding an attribute for an element of the *unconstrained type* must be done using the attribute form. Simply type the name and value of the attribute in the attribute form and click on the  OK button (or press **Enter** in the value field).

The type of an element is unconstrained if

- the element is contained in a document which is not constrained by a grammar,
- OR the content of the element is invalid according to the grammar used to constrain the document.

In both cases, XXE allows the user to add or remove any attribute, the value allowed for these attributes being any string.

In the latter case, fixing the invalid element will automatically make it recover its actual, constrained, type.

- Adding an attribute that matches an *attribute wildcard* must be done using the attribute form. Simply type the name and value of the attribute using the attribute form and click on the  OK button (or press **Enter** in the value field). XXE will of course check that the name you have typed conforms to the attribute wildcard.

Using a W3C XML Schema or a RELAX NG schema (but not a DTD), it is possible to specify an element type which, for example, has 3 "ordinary" attributes `a1`, `a2`, `a3` but also any number of other attributes if the name of these extra attributes belong to certain namespaces, for example, the "`http://www.w3.org/1999/xlink`" namespace.

The set of allowed namespaces is called an attribute wildcard.

4.2.1. Using the attribute table

The content of the attribute table can be described as follows:

- All attributes set for the selected element are displayed by the table.
- All possible attributes for the selected element, *even those not set*, are also displayed by the table.

Attributes which have not been set are displayed in gray. Attributes which have been set are displayed in black.

- Attributes are listed sorted in alphabetical order.
- The names of required attributes are displayed using a bold font.
- The names of fixed attributes are displayed using an italic font.

Procedure for adding an attribute or changing its value:

1. Click on the field at the right of the attribute name. This field is editable.
2. Type the value of the attribute.

If the attribute type is enumerated, this field is a menu rather than a text field, so you can directly choose the value of the attribute without having to type anything.

3. Press **Enter** to commit the change.

Procedure for removing an attribute:

1. Right-click on the row of the attribute to be removed.
2. Select the Remove item in the displayed popup menu.

4.2.2. Using the attribute form

Procedure for adding an attribute or changing its value:

1. Type the name of the attribute in the name field (first field of the form).
2. Press **Enter** to move to the value field (second field of the form).
3. Type the value of the attribute in the value field.
4. Press **Enter** in the value field to commit the change. This also gives the keyboard focus back to the document view.

Tip

Both the name and value fields support auto-completion. However auto-completion in the value field only works for attributes having the following types: any enumerated type, ID, IDREF, IDREFS.

This auto-completion feature can be configured using the Options dialog box.

Procedure for removing an attribute:

1. Select the attribute by either clicking on it in the attribute table or by typing its name in the name field (first field of the form).
2. Press **Enter** to move to the value field (second field of the form).
3. Click on the  Remove attribute button.

Procedure for adding an attribute to an element of the unconstrained type:

1. Type any attribute name in the name field (first field of the form).

Note

If the namespace of the attribute name has not been declared, you will have to declare it using Tools|Declare Namespace before being allowed to add the attribute.

2. Type the value of the attribute in the value field (second field of the form).
3. Press **Enter** in the value field to commit the change. This also gives the keyboard focus back to the document view.

Procedure for adding an attribute that matches an attribute wildcard:

1. Type an attribute name that matches the wildcard in the name field (first field of the form).

Note

If the namespace of the attribute name has not been declared, you will have to declare it using Tools|Declare Namespace before being allowed to add the attribute.

2. Type the value of the attribute in the value field (second field of the form).
3. Press **Enter** in the value field to commit the change. This also gives the keyboard focus back to the document view.

XXE will of course check that the name you have typed conforms to the attribute wildcard.

The attribute form contains the following buttons:

✚ Remove attribute

Removes selected attribute if this attribute is not required.

Clicking on this button with the right mouse button displays a popup menu which allows the user to force the removal of a required attribute.

Using such command is sometimes needed for RELAX NG grammars specifying content models with an elaborate mix of attributes and child elements.

✔ Add or change attribute

Commits the changes. Analogous to pressing **Enter** in the value field.

✘ Cancel

Cancels the non-committed changes and clears the attribute form. Analogous to typing **Esc** in the name or value field.

☰ List of Values

Displays a dialog box which can be used to choose a value for the attribute being edited. The value specified using this dialog box is directly assigned to the attribute.

Specialized helper dialog boxes are displayed for attributes of

- any enumerated type,
- ID, IDREF, IDREFS types,
- hexBinary and base64Binary types.

For all other types of attributes, a very simple multi-line editor is displayed. This may be useful to view or edit attribute values which are too long to be comfortably viewed or edited using the value field.

Note that in the case of the dialog box displayed for enumerated types and ID or IDREF types, a *single click* is sufficient to select a value from the displayed list. Also note that the text field above the list supports *auto-completion*.

📁 Browse Files

Displays a file chooser dialog box. The URL or file name specified using this dialog box is directly assigned to the attribute.

Clicking on this button with the right mouse button displays a popup menu with eight radio buttons allowing to choose between:

- Open or save files.
- File or directories.
- Absolute or relative paths (relative paths are relative to the location of the element being edited).
- URL or plain file names.

The choices made using these radio buttons are persistent during the editing session (but not across editing sessions like, for example, user preferences specified using the Options dialog box).

4.3. Search tool

Procedure for performing a text search:

1. Type text to be searched in Search field or choose it from the menu of the combobox.
2. Change the search options as needed (see below).

3. Type Enter in the Search field or click on the Start button.
4. During the search session
 - click Skip on to skip found text,
 - OR click on Skip Element to skip the element containing the found text.
5. The search session can be stopped by
 - clicking on Stop,
 - OR typing **Esc** (this also gives back the keyboard focus to the document view),
 - OR by simply clicking anywhere in the document view.

Procedure for performing a text search/replace:

1. Type text to be searched in Search field or choose it from combo box menu.
2. Check the Replace toggle
3. Type replacement text in Replace field or choose it from combo box menu.
4. Change the search options as needed (see below).
5. Type Enter in the Search or Replace fields or click on the Start button.
6. During the search/replace session
 - click Skip on to skip found text,
 - OR click on Skip Element to skip the element containing the found text.
 - OR click Replace on to replace found text.
 - OR click Replace All on to replace all occurrences of searched text.
7. The search/replace session can be stopped by
 - clicking on Stop,
 - OR typing **Esc** (this also gives the keyboard focus back to the document view),
 - OR by simply clicking anywhere in the document view.

Search options:

Ignore case

The search is case-insensitive. Example: "foo" matches both "foo" and "Foo".

Whole word

The found string must be a word, that is, the found string must be surrounded by white spaces. Example: "foo" matches "foo" but not "foobar".

Regular expression

The searched string must be a valid regular expression. A regular expression is specified in a syntax similar to that used by Perl. See also <http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html>.

In such case, \$1, \$2, ..., \$9 may be used in the replacement string to refer to the substrings matching the parenthesized groups of the regular expression.

\$0 is replaced by the string matching the regular expression in its entirety. "\$\$" may be used to quote character '\$'.

Examples:

- "f(o+)" matching "foo", replaced by "g\$1", gives "goo".
- "f(o+)" matching "foo", replaced by "\$0bar", gives "foobar".
- "f(o+)" matching "foo", replaced by "g\$\$1", gives "g\$1".

Smart text boundaries

Searching string "Hello world!" in an XML document is not as obvious as it seems: for example, is "Hello world!" with word "Hello" contained in an emphasis element followed by text node " world!" supposed to be found by XXE?

- If this toggle is checked, the answer is yes. "Hello world!" is found within "Hello world!" but not within "<p>Hello </p><p>world!</p>".

This mode uses the grammar constraining current document to recognize logically contiguous text across different types of elements.

- If this toggle is not checked, the answer is no. Each text node is separated from other text nodes whatever the type of the element containing it.

Direction

Up

The document is searched from caret position to the beginning of the document.

Down

The document is searched from caret position to the end of the document.

4.4. Spell tool

Current document is always checked for spelling errors from caret position to its end.

Procedure for checking current document for spelling errors:

1. Click on the Start button or use Tools|Check Spelling.
2. During the check spelling session, use any of the buttons described below.
3. The check spelling session can be stopped by
 - clicking on Stop,
 - OR typing **Esc** (this also gives the keyboard focus back to the document view),
 - OR simply clicking anywhere in the document view.

Language

Selects the language of the dictionary used by the spell checker.

It is possible to switch from a language to another at any time. In such case, the spell checker is automatically restarted with the other dictionary, beginning at the last word for which the spell checker has found an error.

The last selected language is recorded in the user preference file in order to be automatically chosen in subsequent XXE sessions.

Replace

Replaces the erroneous word by the content of the Replace with text field.

An empty text field may be used to delete the erroneous word.

Replace All

Replaces all occurrences of the erroneous word by the content of the Replace with text field.

This button is disabled for errors other than "Unknown word" or "Improperly capitalized word".

Ignore

Skips the word for which the spell checker has found an error.

Ignore All

Skips all occurrences of the word for which the spell checker has found an error.

This button is disabled for errors other than "Unknown word" or "Improperly capitalized word".

Skip Element

Skips the element containing the word for which the spell checker has found an error.

Learn

Records the word in the personal dictionary for currently selected language.

This button is disabled for errors other than "Unknown word" or "Improperly capitalized word".

4.5. Character tool

Click on a character to insert it at caret position in current document view.

The form above the character table can be used to select the range displayed by the character table.

The size of this range is 256 characters.

A range is identified by the Unicode code of its first character. This code is displayed in hexadecimal notation (example: "0x2700" for the dingbats range) but can be input in decimal notation (example: "9984") or in octal notation (example: "023400").

4.5.1. The "Favorites" palette

The last item of the combobox found above the character table is called Favorites. It can be used to select a custom palette of up to 256 characters.

This palette is populated by right-clicking on a character and by choosing the "Add to Favorites" item in the popup menu.

This contextual popup menu contains the following entries:

Copy

Copy the character on which the user has right-clicked to the clipboard.

Add to Favorites

Add the character on which the user has right-clicked to the Favorites palette.

Remove from Favorites

Remove the character on which the user has right-clicked from the Favorites palette.

Show Favorites

The quickest way to switch to the Favorites palette.

4.6. Validity tool

This tool displays the list of validity error messages (if any) found by XXE when opening a document or after the last use of

- Tools|Check Validity or Validity State,
- OR File|Save or File|Save As (validity is automatically checked each time a document is saved).

The color of the message reflects the severity of the error. See icons used by Validity State.

Clicking on the number of an error message selects the element where the validity error was found.

If the document is edited after its validity is checked (typically to fix some of the validity errors), clicking on some of the error messages may have no effect because the corresponding erroneous element no longer exists.

5. Status bar

5.1. Validity state

Clicking on this button displays Validity tool, unless no validity errors are found in current document, in which case an OK message is displayed in the status bar.

This button is disabled if current document is not constrained by a grammar.

Tip

Current document validity is automatically checked each time the document is saved, therefore, unless you are fixing an invalid document, you don't really need to explicitly use this button.

The icon contained in this button shows the validity state of current document:

Icon	Meaning
✓	The document is valid and has no semantic errors.
⚠	The document is valid but has semantic warnings. This happens when the document being edited is itself a W3C XML Schema or a RELAX NG schema or when a user-defined <code>documentHook</code> reports such warnings (See <i>XMLmind XML Editor - Configuration and Deployment</i>).
❌	The document is valid but has semantic errors. This happens when the document being edited is itself a W3C XML Schema or a RELAX NG schema or when a user-defined <code>documentHook</code> reports such errors (See <i>XMLmind XML Editor - Configuration and Deployment</i>).
⚠	The document has minor validity errors such as bad cross-references. Tip Working on a document which has bad cross-references is very common so you should not be alarmed by this situation.
❌	The document has validity errors signaling bad values for attributes or for elements. Note With a DTD, this can only happen for attribute values because the text contained in an element cannot be constrained by this type of grammar.

Icon	Meaning
	<p>This document has severe validity errors such as an invalid child sequence inside an element. <i>It is not a good idea to keep working on a document with such severe validity errors because XXE has not been designed to be convenient to use in such situation.</i></p> <p>Tip</p> <p>After fixing some structure errors using XXE, save and then reload the document.</p> <p>This will automatically discard superfluous white space which was not detected at first because of the validity errors. This trick is much faster than trying to remove invalid white space text nodes by hand.</p>

5.2. Status messages

≡↑ Show Message Log

Displays a dialog box containing last messages reported by XXE.

These messages are sorted by category. One of the most useful category is "Command Execution" which contains the messages reported during the execution of the last command. Commands such as DocBook|Convert Document|Convert to PDF are really verbose, that's why this category is so useful.

Tip

It is also possible to display the same dialog box by right-clicking in the message area of the status bar.

5.3. Clipboard utilities

Clicking on this button displays a dialog box showing the exact content of system clipboard.

Clicking on this button with the right mouse button displays a popup menu with the following items:

View Clipboard Content

Displays a dialog box showing the exact content of system clipboard.

Copy File to Clipboard

Displays standard File Open dialog box. This dialog box can be used to specify the location of an existing plain text or XML file. The content of the specified file is copied to the system clipboard.

Edit|Paste Before, Edit|Paste, Edit|Paste After can then be used to paste the content of the clipboard at locations where the grammar constraining current document allows to do so.

Save Clipboard Content to File

Displays standard File Save dialog box. This dialog box can be used to specify the location of a new plain text or XML file. The content of the system clipboard is saved to the specified file.

If the content of the clipboard consists in one or several XML nodes, this content is saved unindented and using a private namespace which somewhat limits the usability of the saved file.

5.3.1. Clipboard content

Next to the above button, a read-only text field permanently shows a short description of current clipboard content:

- an element name if a single element has been copied or cut to the clipboard,
- OR #text for a text node or for a sequence of characters,
- OR #comment for a comment node.

- OR `#processing-instruction` for a processing instruction node,
- OR the number of nodes (between square brackets) if more than one node have been copied or cut to the clipboard. Example "[5]" if the five list items of this itemized list have been copied to the clipboard.

Note

Once you have copied something to the clipboard from an *external* application, you have to change the "editing context" of XXE if you want to make XXE see what has been copied to the clipboard.

That is:

1. Copy some text to the clipboard from an external application.
2. Click on an element other than the one where you want to paste some text.

Editing context changes and therefore, description `#text` appears in the read-only text field described here.

3. Click inside the element where you want to paste some text.
4. Use Edit|Paste or type Ctrl-V.

6. Dialog boxes

6.1. The "Options" dialog box

6.1.1. Open options

Use URL chooser rather than file chooser

If this toggle is checked, XXE will display an URL chooser dialog box rather than the standard file chooser dialog box each time the user needs to specify a file name.

Default: not checked.

Preference key: `useURLChooser`; type: `boolean`; default: `false`.

Always use the URL chooser to save a copy

If this toggle is checked, XXE will display an URL chooser dialog box rather than the standard file chooser dialog box when the user needs to specify a file name for command Save Copy. That is, the URL chooser dialog box is always used for that command no matter whether toggle "Use URL chooser rather than file chooser" is checked or not.

Default: checked.

Preference key: `useURLChooserToSaveCopy`; type: `boolean`; default: `true`.

When no DTD or schema, guess ignorable white space

If this toggle is checked, when opening a document not constrained by a DTD or a schema, white space characters are stripped from elements containing child elements separated by white space.

This heuristic is generally a good one for XML data.

Default: checked.

Preference key: `guessIgnorableWhitespace`; type: `boolean`; default: `true`.

When no DTD or schema, simulate a document type

If this toggle is checked, a dynamic, non constraining, pseudo document type is used for documents not constrained by a DTD or XML Schema.

The pseudo document type simply remembers all attributes and child elements added to each element during the editing session. Thanks to this pseudo document type, the pick lists of the Edit and Attributes tools are always filled with sensible values.

Default: checked.

Preference key: `useHelperDocumentType`; type: boolean; default: `true`.

Ignore `<?xml-stylesheet?>`

If this toggle is checked, `xml-stylesheet` processing instructions specifying which style sheets to use for the newly loaded document are ignored. If the XXE configuration associated to the newly loaded document specifies CSS style sheets, these style sheets are used instead.

Using this option is useful if `<?xml-stylesheet?>` specifies a style sheet for use by a Web browser such as Mozilla and not a style sheet for use by XXE.

Default: not checked.

Preference key: `ignoreStyleSheetPI`; type: boolean; default: `false`.

Warn about loss of physical structure

If this toggle is checked, a dialog box with a warning is displayed if a document containing *non-managed* references to external entities or non-managed `xi:include` elements is opened in XXE. See managed and non-managed references to external documents.

Default: checked.

Preference key: `warnAboutLossOfPhysicalStructure`; type: boolean; default: `true`.

Inform about non-editable document parts

If this toggle is checked, a dialog box with an information is displayed if a document containing *managed* references to external entities and/or managed `xi:include` elements is opened in XXE. See managed and non-managed references to external documents.

Default: checked.

Preference key: `infoAboutReadOnlyInclusions`; type: boolean; default: `true`.

Managed and non-managed references to external documents

Currently, XXE only knows how to preserve the physical structure of:

- references to external entities containing a *single element*,
- `xi:include` elements pointing to a *single element*.

These references are said to be *managed*. In the document view, these references are replaced by the elements they specify and the referenced elements are *marked as being read-only*.

Tip

Non-editable elements are displayed with a light-gray background (this is specified in the CSS style sheet using the `:read-only` extension pseudo-class -- see *XMLmind XML Editor - Support of Cascading Style Sheets (W3C CSS)*).

Being inside a non-editable node is also clearly indicated by the node path bar.

Tip

See [Edit|Document Reference|Edit Referenced Document](#) to learn how to conveniently edit the documents specified by such managed references.

Preserving the physical structure means that when you modify a document containing such managed references and you save it to disk, XXE will generate a file containing the proper `&entity_name;` and `<xi:include href="location" xpointer="XXX"/>`.

Currently, XXE does not know to preserve the physical structure of:

- references to external entities containing multiple elements,
- references to external entities containing a single element when this element contains references to internal or external entities,
- references to external entities containing a mix of text and elements,
- `xi:include` elements with `parse="text"`,
- `xi:include` elements pointing to the root element plus sibling comments and processing instructions,
- `xi:include` elements pointing to a single element when this element contains references to internal or external entities,
- An `xi:include` element referencing the document containing the include element (that is, `<xi:include xpointer="XXX"/>`).
- etc.

These references are said to be *non-managed*. In the document view, these references are replaced by the nodes they specify and the nature of the reference (external entity or `xi:include`) is definitely lost.

This means that when you modify a document containing such non-managed references and you save it to disk, XXE will generate a "flat" XML file.

6.1.1.1. WebDAV options

Lock

Specifies the type of lock used for the document being edited, when this document is stored on a WebDAV server (supporting locking).

Default: Exclusive

Preference key: `davLockScope`; type: `shared`, `exclusive` or `none`; default: `exclusive`.

Identifier as a lock owner

Specifies a free form string identifying you as an author of documents stored on a WebDAV server. This string is shown to other authors wishing to know who is locking documents on a WebDAV server. It is recommended to specify your email address (example: `john@acme.com`).

Default: user name (example: `john`)

Preference key: `davLockOwner`; type: `string`; default: login name.

Table 1. Lock Types

Lock type	Description
None	The document being edited is not locked.
Shared	The document being edited is locked using a <i>shared</i> lock. A shared lock does not prevent you from overwriting a document shared-locked by another author but at least, you are informed that the document is being "used" by this other author. (In such case, you'll generally want to contact him to decide what to do.)
Exclusive	The document being edited is locked using an <i>exclusive</i> lock. An exclusive lock prevents you from overwriting a document exclusively-locked by another author.

6.1.1.2. Schema cache options

When an instance document conforming to a grammar (DTD, W3C XML Schema or RELAX NG Schema) is opened in XXE, the grammar is checked for validity, then the instance document is checked for validity.

Checking a grammar for validity may be a lengthy operation for a medium-size W3C XML schema and for a large DTD such as DocBook.

Therefore, if the option is enabled, XXE checks the grammar the first time it is used by an instance document and then caches it for subsequent uses. The grammar is cached

1. in memory,
2. on disk using a fast loading binary format (DTD, W3C XML Schema but not RELAX NG Schema).

The directory used to store the fast loading binary files is:

- `$HOME/.xxe/ser/` on Unix,
- `%SystemDrive%\Documents and Settings\%USERNAME%\Application Data\XMLmind\XMLeditor\ser\` on Windows 2000/XP,
- `%SystemDrive%\winnt\Profiles\%USERNAME%\Application Data\XMLmind\XMLeditor\ser\` on Windows NT.

The cache records the location and date of the source files of the grammar. If these source files are removed or modified, the cache will automatically discard the obsolete cached grammar. Of course, this forces XXE to load an up-to-date grammar from the source files.

Note that the existence and date of remote source files (that is, grammar files located on a HTTP or FTP server) are *not* checked by the schema cache.

Enable cache

Enables the behavior described above.

Disabling this option is a good idea if you exclusively use small or medium size DTDs such as XHTML.

Default: checked.

Preference key: `useDocTypeCache`; type: boolean; default: `true`.

Clear cache

Clears the cache forcing XXE to reload a grammar from its source files and to revalidate it next time it is used by an instance document.

This option is useful if you are developing grammars and using XXE to check them.

Note that the cache is automatically cleared each time you install a new version of XXE.

6.1.2. Save options

Encoding

Specifies the encoding used for XML files saved by XXE (if save options have not been specified in an XXE configuration file -- see below).

Default: UTF-8.

UTF-8 is the default encoding of the XML files created by XXE because this encoding is compact and may be used to represent all Unicode characters.

However, if you need to deliver a XML file created by XXE to a person using a text editor to view or modify it, it is nicer to use a human-readable encoding such as ISO-8859-1 (Western character set).

Preference key: `encoding`; type: any encoding supported by XXE; default: UTF-8.

Save characters outside encoding as entity references

If this toggle is checked, all characters not supported by the encoding are saved as entity references.

Example: the Euro sign is not supported by the ISO-8859-1 encoding. If this toggle is checked, the Euro sign is saved as `€`. If this toggle is not checked, the Euro sign is saved as `€`.

Of course, for a character to be saved as an entity reference, the corresponding entity must have been defined in the DTD.

Example: the Euro sign is not supported by the ISO-8859-1 encoding. If there is no entity defined for this currency symbol, the Euro sign is saved as `€` whether the toggle is checked or not.

Default: checked.

Preference key: `saveCharsAsEntityRefs`; type: boolean; default: `true`.

Always save these characters as entity references

Specifies which characters, even if they are supported by the encoding, are always saved as entity references.

Example: the Copyright sign is supported by the ISO-8859-1 encoding but you may prefer to see it saved as `©`. In such case, specify "169" in this text field.

This attribute contains a list of character ranges. A character range is either a single character or an actual range `char1:char2`.

A character may be specified using its Unicode character number, in decimal (example: 233 for e acute), in hexadecimal (example: 0xE9) or in octal (example: 0351).

Because names are easier to remember than numbers, a character may also be specified using its entity name as defined in the DocBook 4.2 DTD (example: `eacute`). Note these symbolic specifications are supported whatever is the DTD or schema of the document being saved.

Examples: `reg 174 0x00ae 0256 pound:yen 163:165 0xA3:0xA5 0243:0245`

Default: empty.

Ignored if toggle "Save characters outside encoding as entity references" is not checked.

Note

There is no need to specify the non-breaking space character (`nbsp` = 160 = 0xa0 = 0240) as it is always saved as ` ` or as ` `.

Preference key: `charsSavedAsEntityRefs`; type: specification as described above; default: empty string.

Indent

If this toggle is checked, XML files saved by XXE are indented (if save options have not been specified in an XXE configuration file -- see below).

The fields and toggles that follows in the Indent frame may be used to parametrize indentation.

Default: checked.

Indentation

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

Default: 2.

Preference key: `indent`; type: integer (negative means not indented); default: 2.

Max. line length

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

Default: 78.

This value is only used as a hint: XML files created by XXE may contain lines much longer than the specified length.

Preference key: `maxLineLength`; type: positive integer; default: 78.

Add open lines

If this toggle is checked, an open line is added between the child elements of a parent element (if the content model of the parent only allows child elements).

Default: checked.

Preference key: `addOpenLines`; type: boolean; default: `true`.

Favor interoperability with HTML and SGML

If this toggle is checked:

- Empty elements having a non empty content are saved as `<tag></tag>`.
- Empty elements having an empty content are saved as `<tag />` (with a space after the tag).

Default: checked.

Table 2. XHTML examples

Toggle checked	Toggle not checked
<p></p>	<p />

Preference key: `favorInteroperability`; type: boolean; default: `true`.

Do not indent unconstrained documents

If this toggle is checked, XML files generated by XXE when saving documents not constrained by a DTD, W3C XML Schema or RELAX NG schema, are not indented, even if the Indent toggle is checked.

Note that when this toggle is not checked, XXE uses very simple heuristics to indent unconstrained documents. Indenting such documents this way may add white space to places where it is significant.

Default: not checked.

Preference key: `dontIndentUnconstrainedDocs`; type: boolean; default: `false`.

Override settings specified in config. files

The above options can also be specified in a configuration file customizing XXE for a specific XML application.

If this is the case, when the configuration is in use, *what has been specified in the Options dialog box is completely ignored*, unless this toggle is checked.

Default: not checked.

Preference key: `overrideConfiguration`; type: boolean; default: `false`.

Automatically show Validity tool

Document validity is automatically checked each time the document is saved to disk. If this toggle is checked, when validity errors are found, the Validity tool ``tab" is automatically selected in order to display the error messages.

Default: not checked.

Preference key: `showValidityPaneOnSave`; type: boolean; default: `false`.

Automatically save modified documents

If this toggle is checked, XXE will automatically save modified documents.

Default: not checked.

Max. modifications before saving

A modified document will be automatically saved after specified number of modifications. Typing a character counts as a modification.

Default: 300.

Preference key: `autoSaveInterval`; type: -1000000-1000000 (number of operations; negative means disabled); default: -300.

Idle time (seconds) before saving

If a modified document is no longer edited during specified number of seconds, this document is automatically saved, even if the "Max. modifications before saving" count has not yet been reached.

This option allows to make the auto-save feature as non-intrusive as possible.

Default: 30.

Preference key: `autoSaveTimeout`; type: 10-86400 (number of seconds); default: 30.

Before saving, make a backup copy of the file

If this toggle is checked, a copy of the original file is made (same name but ending with '~') before saving the modified copy.

Note that backup files are created at the beginning of the editing session and not each time a file is saved. This makes the backup files much more useful. For example a diff between `foo.xml~` and `foo.xml` will show you what you did during last editing session.

Default: checked.

Preference key: `makeBackupFiles`; type: boolean; default: `true`.

6.1.3. Print options

Note

The options for the page footer are not described here because they are identical to those used for the page header (described below).

Screen resolution

Specifies the screen resolution in DPI (Dot Per Inch) used when printing. This resolution directly determines the amount of text a printed page can contain.

Default: 100dpi.

Preference key: `screenResolutionWhenPrinting`; type: 50-100; default: 100.

Begin

The page header has 3 areas: begin (the left for left-to-right languages), middle, end (right). This field specifies the text printed at the left of the page header.

Default: empty.

Each area can contain a mix of text and variables

Preference key: `headerBegin` [`footerBegin`]; type: text; default: empty [%F].

Middle

Specifies the text printed at the center of the page header.

Default: empty.

Preference key: `headerMiddle` [`footerMiddle`]; type: text; default: empty [empty].

End

Specifies the text printed at the right of the page header.

Default: empty.

Preference key: `headerEnd` [`footerEnd`]; type: text; default: empty [%P].

Color

Specifies the color of the text of the page header.

Default: gray.

Note that the font used for the page header is the default font of the style sheet (see the Style options below).

Preference key: `headerColor` [`footerColor`]; type: 3 0-255 integers separated by spaces (specify red, green, blue); default: 128 128 128 [128 128 128].

Underline [Overline]

Specifies if a thin line is to be printed below the page header [above the page footer].

Default: checked.

Preference key: `underlineHeader` [`overlineFooter`]; type: boolean; default: `true`.

Table 3. Substituted Variables

Variable	Description
<code>%F</code>	File name of the document being edited
<code>%f</code>	Same as <code>%F</code> but shortened to approximately 30 characters
<code>%B</code>	Base name of the document being edited
<code>%D</code>	Current date
<code>%T</code>	Current time
<code>%P</code>	Equivalent to localized "page %I of %C"
<code>%I</code>	Current page number
<code>%C</code>	Total page count

6.1.4. Edit options

Clicking with middle button pastes system selection

If this toggle is checked, clicking with the middle button (or with the mouse wheel) pastes the characters copied to the "system selection".

On platforms not supporting system selection (all but generic Unix/Linux), this action pastes the content of an internal clipboard.

Default: not checked.

Preference key: `button2PastesSystemSelection`; type: boolean; default: `false`.

Add interactive margins to styled views

If this toggle is checked, the editor adds gray margins at the left and at the right of the styled document views. These margins allows to trigger special, customizable, actions when the user clicks in them. By default, clicking in these margins, selects the "block" (paragraph, row, row group, table) in front of the click location. Clicking again without moving the mouse, selects the parent of the selected element. Clicking again without moving the mouse, selects the grand-parent, and so on.

Do not click several times too fast otherwise the editor will think you are double-clicking or triple-clicking and therefore, selecting elements that way would not work.

Default: not checked.

Preference key: `addInteractiveMargins`; type: boolean; default: `false`.

Use dialog box rather than Edit tab

Allows to use the Insert dialog box rather than the Edit tab each time an element is to be inserted in the document. This also applies to the replace and convert operations.

Default: not checked.

Preference key: `selectFieldUsingChooser`; type: boolean; default: `false`.

Append mode

Specifies how to quickly complete what is being typed in an autocompletion-enabled text field.

None

There is no quick way to complete what is being typed (other than using Up and Down arrows).

Automatic

What is being typed is automatically completed to compose first possible suggestion.

Manual

The user needs to explicitly press on the space bar (or on **Ctrl-Space**, if some of the choices contain whitespaces) to complete as much as possible what has been typed.

Example: the list of choices is "aaz aaa bbz bbb ccz ccc". The user types "b".

None

Nothing happens and pressing on the space bar has no effect.

Automatic

The text field is automatically updated to contain "bbz".

Manual

Nothing happens, but pressing on the space bar will add a "b" to what has been typed (which gives "bb").

Default: Manual.

Preference key: `appendSuggestionMode` ; type: none, auto or manual; default: manual.

In the choice list, only show suggestions

If this toggle is checked, the list which displays possible choices (e.g. elements to be inserted in the document) is kept as short as possible.

Default: not checked.

By default, such list displays all possible choices, whether matching what has been typed in the autocompletion-enabled text field or not.

Example: the list of choices is "aaz aaa bbz bbb ccz ccc". The user types "b".

If this toggle is not checked, the list contains "aaz aaa bbz bbb ccz ccc" and item "bbz" is highlighted.

If this toggle is checked, the list contains "bbz bbb" and item "bbz" is highlighted.

Preference key: `onlyShowSuggestions`; type: boolean; default: false.

Max. undo actions

Specifies the maximum number of undo (redo) actions a user will be able to perform. Limited to 100 because a single undo action may consume a great deal of memory.

Default: 20.

Preference key: `maxUndo`; type: 1-100; default: 20.

Type of document reference

Specifies the type of document/element reference pasted after using command Edit|Document Reference|Copy as Reference.

External entity

The Paste command must always use the "reference to an external entity" mechanism. Choose this option to ease the interchange of the documents you create with XXE.

Important

It is not possible to use the "reference to an external entity" mechanism when the referencing document (example: a book document `book.xml` referencing a chapter document `chap1.xml`) has no document type declaration (`<!DOCTYPE>`).

The reason for this is that XXE needs to declare the referenced entities in the internal DTD subset and that XXE will not create an internal DTD subset unless the referencing document already has a `<!DOCTYPE>`.

XInclude

The Paste command must always use the XInclude mechanism. See <http://www.w3.org/TR/xinclude/>. Choose this option if your documents are constrained by W3C XML Schemas or RELAX NG schemas or if you often need to reference an element inside a document rather than the whole document.

This element more or less deeply inside the referenced document will be specified by XXE using an XPointer using the element scheme. See <http://www.w3.org/TR/xptr/>.

Note that you do not need to declare `xi:include` elements in the DTD or in schema because XInclude processing occurs before validation. That is, `xi:include` elements are already replaced by the included elements when validation begins.

Entity when possible, XInclude otherwise

The Paste command must use the "reference to an external entity" mechanism when possible (example: the referencing document has a `<!DOCTYPE>` and no XPointer is being used) and the XInclude mechanism otherwise (example: the referencing document does not have `<!DOCTYPE>` or an XPointer is needed).

This is the recommended option.

Default: Entity when possible, XInclude otherwise.

Preference key: `inclusionType`; type: `entity`, `xinclude` or `auto`; default: `auto`.

Warn about hierarchical XPointers

Command Edit|Document Reference|Copy as Reference allows to copy as a reference *any element*, including those not having an ID.

Unless the copied reference is a reference to the root element of a document, this means that the copied reference is based on the position of the element as a child of its parent element, on the position of the parent element as a child of the grand-parent element and so on. This is what we call a *hierarchical XPointer*. Example: `file:///usr/share/docs/xxe/doc/docbook/help.xml#element(/1/3)`.

Pasting a reference based on a hierarchical XPointer is *not safe* because if you modify the referenced document, your referencing document may point to an element other than the one for which a reference has been originally pasted.

This is why it is strongly recommended not to uncheck this option, which warns you each time you try to paste a reference based on a hierarchical XPointer.

Note that you'll never get a warning dialog box when you'll paste a reference to the root element of a document (example: `file:///usr/share/docs/xxe/doc/docbook/help.xml`) or when you'll paste a reference which exclusively uses an ID to locate the element (example: `file:///usr/share/docs/xxe/doc/docbook/help.xml#element(parameterGroup)`) because these references are considered to be safe.

Default: checked.

Preference key: `warnAboutHierarchicalXPointer`; type: `boolean`; default: `true`.

6.1.5. View options

The following options parameterize the CSS style sheet used to visualize the document or to print it.

Setting some of these preferences will have no visible effect if the style sheet author has specified the corresponding properties in the style sheet. For example, if the user's preferred background is specified in the Style section as being light yellow and if the style sheet author has specified the root element `background-color` as being white, the document will be rendered with a white background.

Serif font family

Specifies the font family used for property value `font-family:serif`.

Default: Serif (the Java™ default serif font family).

Preference key: `serifFontFamily`; type: font family name; default: `Serif`.

SansSerif font family

Specifies the font family used for property value `font-family:sans-serif`.

Default: SansSerif (the Java™ default sans-serif font family).

Preference key: `sansSerifFontFamily`; type: font family name; default: `SansSerif`.

Monospaced font family

Specifies the font family used for property value `font-family:monospace`.

Default: Monospaced (the Java™ default monospaced font family).

Preference key: `monospaceFontFamily`; type: font family name; default: `Monospaced`.

Default font family

Specifies the default value for property `font-family`.

Default: `SansSerif`.

Preference key: `defaultFontIsSerif`; type: boolean; default: `false`.

Default font size

Specifies the default value for property `font-size`.

Default: 12pt.

Preference key: `defaultFontSize`; type: 10-20 (pt); default: 12.

Default background

Specifies the default value for property `background-color`. (Displays standard color chooser dialog box.)

Default: white.

Preference key: `defaultBackground`; type: 3 0-255 integers separated by spaces (specify red, green, blue); default: 255 255 255.

Default text color

Specifies the default value for property `color`. (Displays standard color chooser dialog box.)

Default: black.

Preference key: `defaultForeground`; type: 3 0-255 integers separated by spaces (specify red, green, blue); default: 0 0 0.

6.1.6. Tools options

6.1.6.1. Spell options

Ignore case

If this toggle is checked, ignore capitalization errors.

Default: not checked.

Preference key: `ignoreCase`; type: `boolean`; default: `false`.

Ignore mixed-case words

If this toggle is checked, do not check words containing case mixing (e.g. "SpellChecker").

Default: not checked.

Preference key: `ignoreMixedCase`; type: `boolean`; default: `false`.

Ignore words with digits

If this toggle is checked, do not check words containing digits (e.g. "b2b").

Default: checked.

Preference key: `ignoreDigits`; type: `boolean`; default: `true`.

Ignore URL-like words

If this toggle is checked, ignore words looking like URLs or file names (e.g. "www.xxx.com" or "c:\boot.ini").

Default: checked.

Preference key: `ignoreURL`; type: `boolean`; default: `true`.

Ignore duplicate words

If this toggle is checked, do not signal two successive identical words as an error.

Default: not checked.

Preference key: `ignoreDuplicates`; type: `boolean`; default: `false`.

Check punctuation

If this toggle is checked, punctuation checking is enabled: misplaced white space and wrong sequences, like a dot following a comma, are detected.

Default: not checked.

Preference key: `checkPunctuation`; type: `boolean`; default: `false`.

Allow compounds words

If this toggle is checked, all words formed by concatenating two legal words with an hyphen are accepted. If the language allows it, two words concatenated without hyphen are also accepted.

Default: checked.

Preference key: `allowCompound`; type: `boolean`; default: `true`.

Allow general prefixes

If this toggle is checked, a word formed by concatenating a registered prefix and a legal word is accepted. For example if "mini-" is a registered prefix, accepts "mini-computer".

Default: checked.

Preference key: `allowPrefixes`; type: `boolean`; default: `true`.

Allow file extensions

If this toggle is checked, accepts any word ending with registered file extensions (e.g. "myfile.txt", "index.html", etc).

Default: checked.

Preference key: `allowFileExtensions`; type: boolean; default: `true`.

Favor quality over speed

For use on a fast computer. Does not influence the number of suggestions (always limited to 15).

Default: not checked.

Preference key: `maxSuggestionForce`; type: boolean; default: `false`.

6.1.6.2. Spreadsheet options

Open spreadsheets in auto-update mode

Specifies that spreadsheets are to be opened in auto-update mode.

In manual update mode, only newly inserted formulas are computed. To force a full calculation, the user has to explicitly use `Tools|Spreadsheet|Update`.

In auto-update mode, a full calculation is automatically performed, if needed to, when the editing context changes. For example: type some text in a paragraph, then click in (or tab to) another paragraph to trigger a spreadsheet calculation.

Note that in both modes, a full calculation is automatically performed, if needed to, before validating or saving the document.

Using manual update mode is recommended if you have a slow computer or if you have inserted a lot of formulas in your document or if your formulas access many external documents.

Default: checked.

Preference key: `spreadsheetAutoRecalc`; type: boolean; default: `true`.

Maximum number of iterations allowed for calculations in spreadsheet

Specifies the maximum number of iterations allowed for calculations in spreadsheet. This limit is used to prevent the spreadsheet engine from looping in case of cycles in formulas.

Default: 20.

Preference key: `spreadsheetMaxIterations`; type: 2-2000; default: 20.

Cache external documents accessed by formulas

Formulas can access external documents using *XPath escapes* (example: ``document("tutorial/VAT-rates.html#france_vat",.)``). Such access is generally slow and therefore, documents need to be cached the first time they are loaded.

However, the cache is not very smart and will often not be able to detect changes in the external document. This problem will happen if the change happens in a module included by the document or if the document is stored on a remote HTTP or FTP server. In this case, disabling the cache and also disabling the auto-update mode are recommended.

Default: checked.

Preference key: `spreadsheetCacheDocuments`; type: boolean; default: `true`.

6.1.7. Window options

Show both tree and styled views

If this toggle is checked, XXE will automatically create two views for a newly opened or newly created document. That is, by default, a document tab will contain a tree view and a styled view side by side.

This option has of course no effect on documents for which no CSS style sheet is available.

Default: not checked.

Preference key: `showBothViews`; type: boolean; default: `false`.

Place tree view at right

If this toggle is checked, the tree view is to be placed at the right of the styled view.

Default: not checked.

Preference key: `treeViewAtRight`; type: boolean; default: `false`.

Tree view width

Specifies the width of the tree view in percentage of the available window area.

Default: 33%.

Preference key: `treeViewPercent`; type: 10-90; default: 33.

6.1.8. General options

Use a unique instance of XMLmind XML Editor

If this toggle is checked, a unique instance of XMLmind XML Editor is used to open all your XML documents. For example, if you double-click on the icon of an XML document in the file ``explorer'', the running instance is used to open this XML document. When this toggle is not checked, a new instance of XXE is started in order to load the document clicked upon.

If there is no running instance of XXE, a new instance is started. If the running instance is hidden by other windows or is iconified, it is made visible before opening the requested document.

This feature also works fine from the command line. For example, on Linux, if this toggle is checked, executing `"xxe mydoc.xml &"` will cause the running instance to be used to open `mydoc.xml`.

If you need to change this option, make sure to close all running instances except one. Then change the option in the last instance and close it. Then restart XXE.

Default: not checked.

Preference key: `singleInstance`; type: boolean; default: `false`.

Port

Specifies the TCP/IP port used by different instances of XXE to communicate with each other. It is highly recommended to use a port in the dynamic/private range: 49152 to 65535.

Default: 49987

Preference key: `singleInstancePort`; type: 1-65535; default: 49987.

Font size

May be used to change the base font size of XXE menus and dialog boxes.

Note that this orthogonal to changing the base font size used to display styled documents.

Default: default base font size of Java™ applications.

Preference key: `fontSize`; type: 10-20 or -1 (means default); default: -1.

Locale

May be used to force the language used in XXE menus and dialog boxes. For example, may be used to force the use of English on a machine where the default locale is German.

Default: default locale of the machine running XXE.

Preference key: `locale`; type: *language* (e.g. `fr`) or *language_COUNTRY* (e.g. `fr_CA`) or - (means default); default: -.

Style

May be used to change the look and feel of XXE user interface.

Default: the system look and feel, except on Linux: Metal when Java™ 1.5 is used to run XXE; Kunststoff when Java™ 1.4 is used to run XXE.

Preference key: `lookAndFeelClassName`; type: Java™ class name of a PLAF or - (means default); default: -.

6.2. The "Declare Namespace" dialog box

Important

XXE is not namespace aware for a document using a DTD as its grammar.

Being not namespace aware means that `xmlns` attributes are treated as ordinary attributes with no special semantics. All element and attributes names (except `xml:space`, `xml:lang`, `xml:base` which belong to the "`http://www.w3.org/XML/1998/namespace`" namespace) have no namespace. Names like "`foo:bar`" are allowed.

A DTD is considered to be a grammar if it defines at least one element type. That is, a DTD only declaring entities is not considered to be a grammar.

This dialog box is not displayable for non-namespace aware documents.

This dialog box can be used

- To declare a namespace used by the name of an attribute or an element (it is required to do so before adding the element or the attribute) or used in the value of the attribute or in the textual content of the element (for example, when the document being edited is a schema).

In practice, this is very rarely needed because document templates generally contain declarations for all the namespaces they might use.

- To change the ``prefix" of a namespace.

What is called a ``prefix" here should be called the *nickname* or the *mnemonic* of the namespace. Without nickname "`xsi`" for namespace "`http://www.w3.org/2001/XMLSchema-instance`", an attribute name such as "`xsi:schemaLocation`" would be displayed as "`{http://www.w3.org/2001/XMLSchema-instance}schemaLocation`" by the GUI of XXE, which is quite unreadable.

Real namespace prefixes only exist while the document is being loaded and while the document is being saved. Do not try to add `xmlns` attributes to elements to specify them. XXE does this automatically each time the document is saved, the nicknames declared using this dialog box being used preferably to automatically generated prefixes such as `ns`, `ns2`, `ns3`, etc.

- To make a namespace the ``default" namespace, that is the namespace for which no prefix is displayed (for element names, not for attribute names).

It is not recommended to mark a namespace as being ``default" if some elements in the document have names without a namespace. In such case, it would be impossible to tell if name "title" is "title" with no namespace or is in fact "{http://www.foo.com/namespace}title", where "http://www.foo.com/namespace" has been marked as being the default namespace.

Procedure for declaring a namespace:

1. You may have to clear the form by clicking on the ✖ Cancel button
2. Specify the URI of the namespace in the Namespace field.
3. Specify the ``prefix" of the namespace in the Prefix field.

Specifying a prefix is mandatory even if the namespace is to be marked as ``default".

4. Optionally mark it as being the `default" namespace by checking the Default toggle.
5. Type Enter in any field or click on the ✔ OK button.

Procedure for changing the declaration of a namespace:

1. Select the namespace declaration to be edited by clicking on its row in the table.
2. Specify the ``prefix" of the namespace in the Prefix field.

Specifying a prefix is mandatory even if the namespace is to be marked as ``default".

3. Optionally mark it as being the ``default" namespace by checking the Default toggle.
4. Type Enter in any field or click on the ✔ OK button.

6.3. The URL chooser dialog box

The URL chooser dialog box may be used to specify the location of a file or directory on the local file system or on a remote FTP or WebDAV server. This file or directory may be an existing one or a file or directory to be created; it depends on the command displaying this dialog box.

An *URL* (Uniform Resource Locator), often called *Internet Address* in Web browsers, has two parts:

- The server part, which is specified in the upper side of the dialog box.
- The file path part, which is specified in the File path text field, located in the lower side of the dialog box.

Procedure for specifying an URL:

1. Choose between the local file system (click on File toggle) and remote FTP or WebDAV servers (click on FTP or HTTP/HTTPS toggles).

Note

A WebDAV server is an HTTP server supporting *WebDAV* (Web-based Distributed Authoring and Versioning, see <http://www.webdav.org/>) extensions.

Note

A *HTTPS* server is an HTTP server using a Secure Socket Layer (SSL). A secure socket layer is a protocol designed to transfer encrypted data between computers over the Web.

When such protocol is used, the server sends a certificate to the client in order to identify itself. This certificate contains the *fully qualified name* of the server, example: `fs23.acme.com`. Make sure to

also specify this fully qualified name in the Server: field (that is, specify `fs23.acme.com`, not simply `fs23`) or you'll get an open error.

HTTPS servers also allow *the client* to send a certificate as a mean to authenticate himself. That is, HTTPS servers support an elegant alternative to the login+password authentication. Unfortunately, this technology is currently not supported by XXE. Your administrator needs to setup the HTTPS server to authenticate users using the standard BASIC or DIGEST algorithms.

2. For a file located on a FTP server, you'll need to choose between an anonymous session (click on the Anonymous toggle) and a session using a named account (click on the User toggle and specify user name and password in the corresponding text fields).

For a file located on a WebDAV server, you'll have to specify your user name and password *after* the connection to the server has been established. In order to do that, a specialized dialog box will be displayed at that time.

3. Specify the file path in the File path text field. The easiest way to do that is to click on the Browse Files button in order to be able to use a specialized file chooser. Which file chooser is displayed depends on what you specified in upper side of the dialog box.

A file path always use '/' as a separator for its components. For example, on Windows, `C:\WINNT\Profile` has to be specified as `/C:/WINNT/Profile`. But you don't have to worry about this if you use the specialized file chooser displayed when you click on the Browse Files button.

Note

When choosing a file located on a WebDAV server, you'll have to specify by hand the WebDAV enabled initial directory (unless the root directory, '/', is WebDAV-enabled on your HTTP server).

On most HTTP server, trying to browse files starting from '/' will fail with error "'/' is not DAV-enabled". In such case, you'll have to enter a path such as `"/dav/mydocs/"` first, then click on the Browse Files button to select a file or directory below this WebDAV-enabled prefix.

Tip

When browsing files on a WebDAV server, icon  shows which files are locked and icon  shows which directories are locked. Moving your mouse over these icons will display a tool tip listing the authors locking the corresponding file or directory.

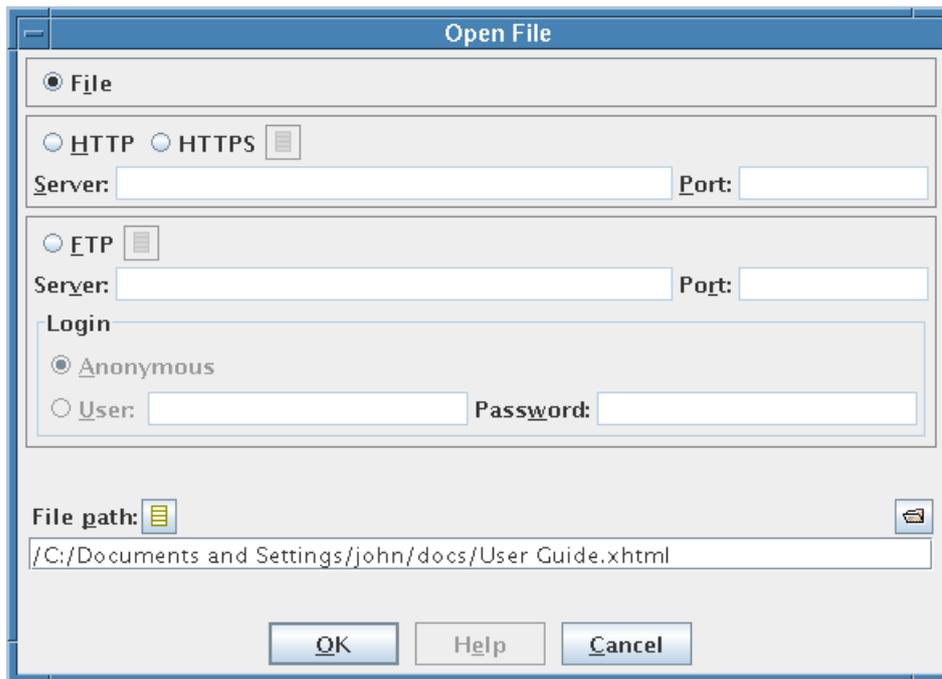
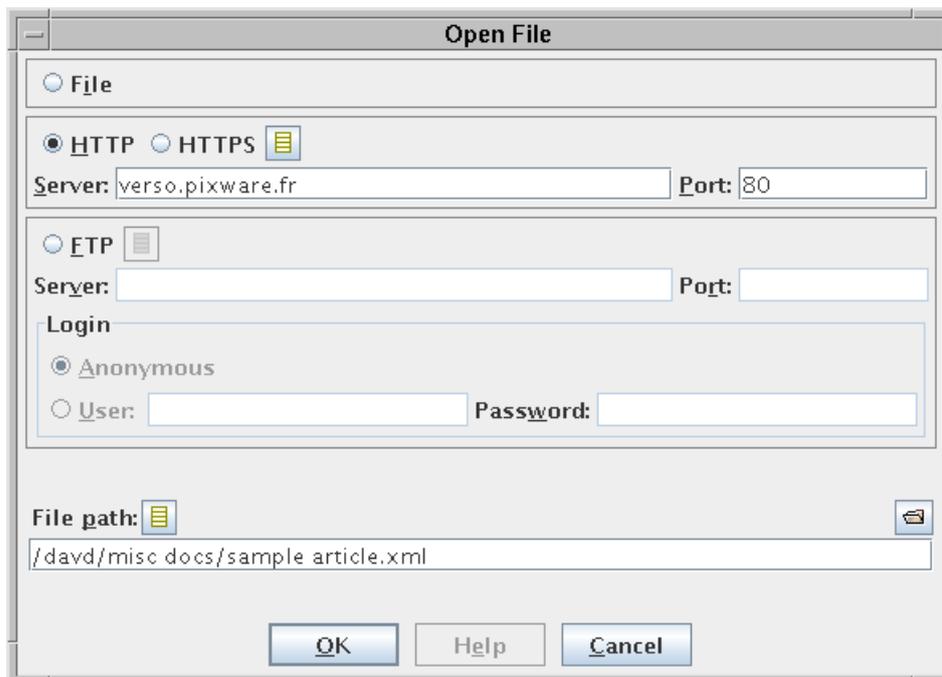
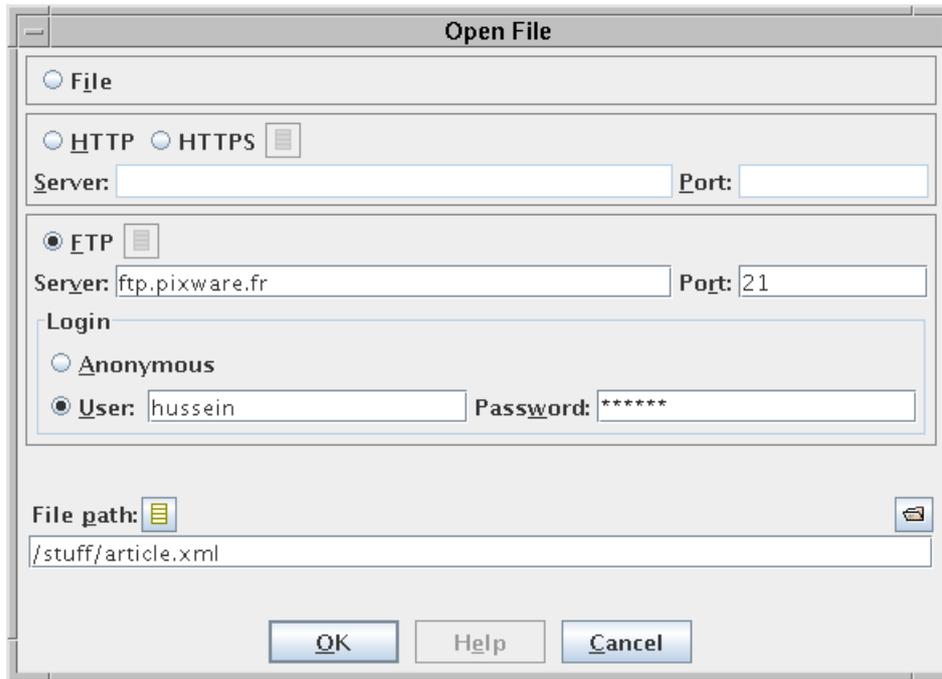
Figure 1. File example on Windows**Figure 2. HTTP example**

Figure 3. FTP example

6.4. The "Resources" dialog box

This dialog box is automatically displayed during a File|Save As operation when the document to be saved has resources which are logically part of this document.

There are 3 types of resources:

- External entities. Displayed in green. Right-clicking on such resource will pop up a dialog box showing all its sub-resources if any.
- XIncludes. Displayed in blue. Right-clicking on such resource will pop up a dialog box showing all its sub-resources if any.
- Application resources (typically graphics files). Displayed in black.

Application resources must be explicitly specified for each document type in the corresponding XXE configuration file using a `documentResources` configuration element. See *XMLmind XML Editor - Configuration and Deployment*.

For each resource, there are 3 possible actions:

Ignore

The reference to the resource is the same in the original and in the saved document.

Example: if document `/docs/docA.xml` having resource `images/icon.png` is to be saved as `/new-docs/docB.xml`, `icon.png` will not be copied to `newdocs/` and `docB.xml` will have the same reference to the resource as `docA.xml`, that is `images/icon.png` (probably not a valid reference).

Copy to

Copy the resource to the location of the saved document. The saved document references its own copy of the resource.

Example: if document `/docs/docA.xml` having resource `images/icon.png` is to be saved as `/new-docs/docB.xml`. The dialog box will suggest to copy `icon.png` to `/newdocs/images/icon.png` in order to

be able to reference its own copy as `images/icon.png`. Note that the `/newdocs/images/` subdirectory will be created on the fly if needed to.

Reference as

Do not copy the resource to the location of the saved document. The saved document references the original resource.

Example: if document `/docs/docA.xml` having resource `images/icon.png` is to be saved as `/newdocs/docB.xml`. The dialog box will suggest to reference `icon.png` as `../docs/images/icon.png`.

6.5. The formula editor

Formula

Type a formula in this text area.

Unlike in most spreadsheet software,

- A formula can contain spaces.
- Use of local variables and intermediate formulas is allowed.
- Comment lines are allowed and must start with '#'.
- Use newlines to separate intermediate formulas and comment lines.

Example:

```
vat = `document("tutorial/VATrates.html#france_vat", .)`
# Trim the '%' sign at the end of vat.
= left(vat, len(vat) - 1)
```

Warning

Do not use names for your local variables that look like cell references. Example: 'x' will work, but not 'x1'. That is, `[a-zA-Z][a-zA-Z]*[1-9][1-9]*` (as well as `true/TRUE/false/FALSE` — the formula language is case insensitive) are reserved identifiers.

Help about selected function

Select a function name in the Formula text area and click on this button to switch to the other tab and display online help about the selected function. Shortcut: F1.

Evaluate selected text

Select an expression in the Formula text area and click on this button to evaluate it and display a dialog box containing the result of this evaluation. Shortcut: F2.

This is handy if you want to experiment with a predefined function you don't really understand.

Evaluate formula

Click on this button to evaluate the whole formula and display a dialog box containing the result of this evaluation. Shortcut: F3.

Format

The format fields are useful to separate calculation from formatting.

Use case: let's suppose your formula computes an amount of money. You need to insert in the document this amount nicely formatted, preceded by string `"Total: "` and followed by string `" (excluding taxes)"`.

Of course, this can be done like this:

```
amount = ...
= "Total: " & numbervalue(amount, "0.00") & " (excluding taxes)"
```

An alternative is to use the Format fields. First field contains a prefix which is prepended to the formatted result. Second field contains a date or number format used to format the raw result. A combobox allows to specify the locale used to interpret this format. Third field contains a suffix which is appended to the formatted result.

Therefore, specify "Total: " in the first field. Choose "#,##0.00" from the combobox. Specify " (excluding taxes)" in the third field.

Change value of attribute

A formula computes a value. This value can be used to add/replace the text node immediately after the formula or this value can be used to add/replace an attribute of the element containing the formula.

If you want to use a formula to change an attribute, check this toggle and use the associated combobox to specify the name of this attribute.

Disabled

Check this toggle to disable the formula being edited.

Disabling a formula means passivating it. That is, it is no longer used to update the document. In some cases, this may be a handy alternative to removing it.

Tip

In the styled view, formulas are represented by a small F icon. Clicking on this icon with the middle button of the mouse allows to switch the state of the formula from enabled  to disabled  and vice-versa.

6.6. The "Find Element" dialog box

This dialog box allows to select nodes specified using an XPath expression. This dialog box has a Simple tab which allows to perform most common search tasks without having to learn XPath. Arbitrarily complex XPath expressions are specified using the Advanced tab.

The search starts at explicitly selected node, if any, and at implicitly selected element otherwise.

6.6.1. The "Simple" tab

Example 1: find element having attribute `id` equals to `introduction`:

1. Select "First in document". Make sure the Element field is empty.
2. Check Having attribute. Type "id" in the text field. Select "Equals to". Type "introduction" in the next text field.
3. Make sure Containing text is not checked.
4. Click OK.

Example 2: navigate from element `para` to next element `para`:

1. Click at the very beginning of your document.
2. Select "Following current node". Type "para" in the Element field.
3. Make sure Having attribute is not checked.
4. Make sure Containing text is not checked.
5. Click OK.

6. Press **Ctrl-A** (command Find Element is repeatable) to move to next `para`, if any.

Example 3: find element `html:pre` having a `class` attribute and containing some text matching regular expression `"print\\w+\\("`:

1. Select "First in document". Type `html:pre` in the Element field.

If the namespace corresponding to prefix "html" is the default namespace of the document, it is also possible to simply type `pre`.

2. Check Having attribute. Type `class` in the text field. Make sure the next text field is empty.
3. Check Containing text. Select "Matching RE". Type `print\\w+\\("` in the text field.
4. Click OK.

6.6.2. The "Advanced" tab

Specify an XPath expression in the corresponding text field.

This expression is evaluated in the context of the explicitly selected node, if any, and in the context of the implicitly selected element otherwise.

The evaluation of the XPath expression must return a *nodeset*. If this nodeset exclusively contains *contiguous siblings*, all the nodes in the nodeset are selected. Otherwise, first node (in document order) of the nodeset is selected.

If the evaluation of the expression returns attributes, the corresponding elements are selected.

It is not possible to select the document node or sibling nodes of the root element.

A. Command line options

```
xxe [-putpref key value]* [-delpref key]* [file_or_URL_to_be_opened]*
```

`-putpref key value`

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

The set of the user's preferences is stored in `XXE_user_preferences_dir/preferences.properties` and is normally modified using the Options dialog box.

The description of each option includes a short description of the corresponding preference key. For example, in the case of the Encoding option, the preference key is `encoding`. Example:

```
xxe -putpref encoding Windows-1252
```

Note

XXE user preferences directory is:

- `$HOME/.xxe/` on Unix,
- `%SystemDrive%\Documents and Settings%\%USERNAME%\Application Data\XMLmind\XMLeditor\` on Windows 2000/XP,
- `%SystemDrive%\winnt\Profiles%\%USERNAME%\Application Data\XMLmind\XMLeditor\` on Windows NT.

`-delpref key`

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