
XMLmind XML Editor - Online Help

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

June 5, 2009

Abstract

This online help contains, in addition to a short tutorial, the reference manual of all the menus, tool bars and dialog boxes of XMLmind XML Editor (XXE for short).

1. Getting started with XMLmind XML Editor	2
2. Menus	9
2.1. File menu	9
2.2. Select menu	10
2.2.1. The selection in XXE	10
2.2.1.1. Text selection	10
2.2.1.2. Node selection	11
2.2.1.3. Implicit element selection	12
2.2.2. Menu items	12
2.3. Edit menu	13
2.3.1. Text menu	17
2.3.2. Comment menu	18
2.3.3. Processing instruction menu	18
2.3.4. Reference menu	18
2.4. Search menu	21
2.4.1. Bookmark menu	21
2.5. View menu	22
2.6. Tools menu	23
2.6.1. Helper Application menu	23
2.6.2. Spreadsheet menu	25
2.6.3. "Record Macro" menu	26
2.7. XML menu	27
2.8. Options menu	27
2.8.1. "Quick Preferences" menu	28
2.8.2. "Customize Configuration" menu	29
2.9. Window menu	30
2.10. Help menu	30
3. The document view area	31
4. Main tool bar	32
5. Node path/node selection bar	33
5.1. Document reference tool bar	33
5.2. Node Path bar	33
5.2.1. Non-editable nodes	34
5.2.2. Elements which are in non-validating mode	34
5.3. Select tool bar	35
6. Tools	35
6.1. Edit tool	35
6.2. Include tool	37
6.2.1. Working with ``XML variables"	37
6.2.2. Using the Include tool	38
6.3. Attributes tool	39
6.3.1. Using the attribute table	40
6.3.2. Using the attribute form	40
6.4. Search tool	42
6.5. Spell tool	44
6.6. Character tool	46

6.6.1. The "Favorites" palette	46
6.7. Validity tool	47
7. Status bar	47
7.1. Automatic Spell Checker	47
7.2. Validity state	47
7.3. Status messages	48
7.4. Overwrite mode	48
7.5. Clipboard utilities	49
7.5.1. Clipboard content	49
8. Dialog boxes	50
8.1. The URL chooser dialog box	50
8.2. The "Find Element" dialog box	52
8.2.1. The "Simple" tab	52
8.2.2. The "Advanced" tab	53
8.3. The "Declare Namespace" dialog box	53
8.4. The spreadsheet formula editor	54
8.5. The "Preferences" dialog box	56
8.5.1. New options	56
8.5.2. Open options	56
8.5.3. Save options	58
8.5.4. Print options	61
8.5.5. Edit options	63
8.5.6. View options	65
8.5.7. Tools options	66
8.5.7.1. Spell options	66
8.5.7.2. Validate options	68
8.5.7.3. Spreadsheet options	68
8.5.7.4. Helper applications options	69
8.5.8. Window options	71
8.5.9. Install add-ons options	71
8.5.10. General options	73
8.5.10.1. Features Options	74
8.5.11. Advanced options	74
8.5.11.1. Schema cache options	74
8.5.11.2. Proxies options	75
8.6. The "Install Add-ons" dialog box	77
8.7. The "Helper Application Editor" dialog box	78
8.8. The "Save Document As Template" dialog box	80
8.8.1. Removing a previously defined document template	81
8.9. The "Save Selected Element as Template" dialog box	81
8.9.1. Advanced options	82
8.9.2. Removing a previously defined element template	83
8.10. The "Add Keyboard Shortcut" dialog box	84
8.10.1. Removing a previously defined keyboard shortcut	87
8.11. The "Change Document Conversion Parameters" dialog box	88
8.11.1. The "Parameter Set Chooser" dialog box	90
A. Command line options	91
1. Command line	91
2. Environment variables	93
3. System properties	94

1. Getting started with XMLmind XML Editor

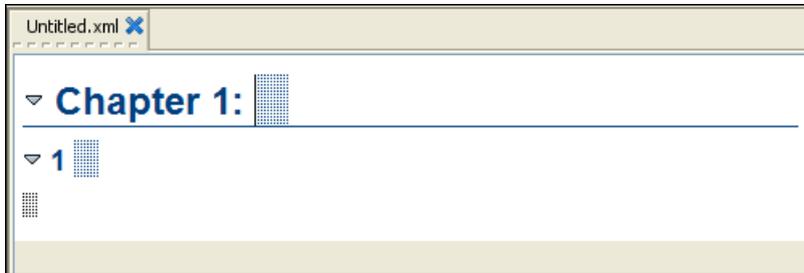
Note

Everything explained here applies not only to DocBook, but also to all the other document types (XHTML, DITA, your custom schema, etc) supported by XMLmind XML Editor.

Creating a document

Use File → New, select DocBook v5+|Chapter, then click OK.

Figure 1. A newly created chapter



The text you see, `Chapter 1`, `1`, etc, is automatically generated and cannot be modified. The square patterns you see are *text placeholders*.

Type the title of your chapter in the first placeholder. Click on (or tab to) second placeholder and type the title of the first section. Do the same for the first paragraph of the first section.

How do you know where you are? Answer: always keep an eye at the *node path bar*.

Figure 2. The node path bar

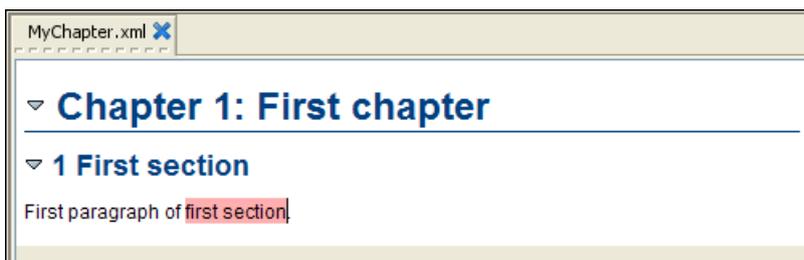


The node path bar shown above indicates that the caret¹ is located inside a `text` node contained in a `para` element, itself contained in a `section` element, itself contained in a `chapter` element.

The text selection

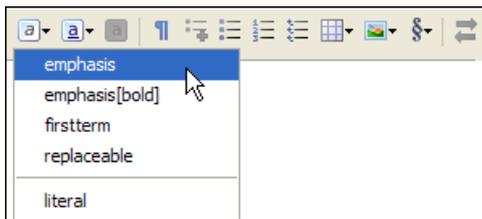
Select some text by dragging the mouse over it, as you would do it in any text editor.

Figure 3. The text selection



You can now convert this text to an `emphasis` element. For that, select the `emphasis` item from the menu of the toolbar button shown below.

¹The caret is also called the insertion cursor.

Figure 4. The "Convert to emphasis" toolbar button

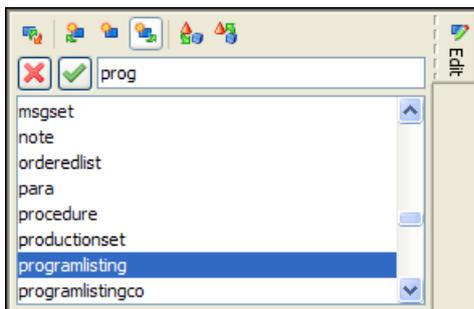
You'll not find all the elements you want in this menu. For example, you'll not find `citetitle` or `uri`. In the general case, you'll have to use the  Edit → Convert (**Ctrl+T**)² command to convert the selection. More on this command later.

The node selection

The implicit element selection

Let's suppose you want to add a `programlisting` element after the `para` element. Click anywhere inside the `para` element, except inside its `emphasis` child element. Even if you don't see anything special about it, this `para` is now *implicitly selected*. This means that all the commands you may invoke will use this `para` as their subject.

Use  Edit → Insert After (**Ctrl+J**). Doing this activate the  *Edit tool*. Type the first few letters of the name of the element you want to insert, then press Enter.

Figure 5. The Edit tool

A `programlisting` element has been inserted after the `para` element. The caret is inside it. You can type some text in it right away.

Note that you have used command Insert After and not command Insert.

Table 1.  Edit tool commands

Command	Shortcut ²	Description
 Replace	Ctrl+R (R like Replace)	Replace the node selection by a newly created element. Example: you want to replace the <code>title</code> child element of a <code>section</code> by the more comprehensive <code>info</code> element. You cannot delete the <code>title</code> child element because the content model of a <code>section</code> specifies that a <code>section</code> must start with a <code>title</code> or an <code>info</code> . However you can <i>replace</i> the <code>title</code> by an <code>info</code> .

²On the Mac, use the Command key instead of the **Ctrl** key.

Command	Shortcut ²	Description
 Insert Before	Ctrl+H (I like Insert; H is before I)	Insert a newly created element before the node selection.
 Insert	Ctrl+I (I like Insert)	Insert a newly created element <i>at caret position</i> .
 Insert After	Ctrl+J (I like Insert; J is after I)	Insert a newly created element after the node selection.
 Convert	Ctrl+T (T like Transform)	If a single element is selected, convert this element to another element which has a compatible content. Example: convert an <code>itemizedlist</code> to an <code>orderedlist</code> . If multiple nodes are selected, wrap a newly created parent element around this nodes. Example: wrap a <code>blockquote</code> around several selected <code>paras</code> .
 Convert [wrap]	Ctrl+Shift+T	A variant of command Convert. Always wraps a newly created parent element around the selected nodes. Example: a single <code>para</code> is selected. Convert allows to convert it to a <code>simpara</code> or a <code>programlisting</code> . Convert [wrap] allows to wrap a <code>blockquote</code> around it.

The explicit node selection

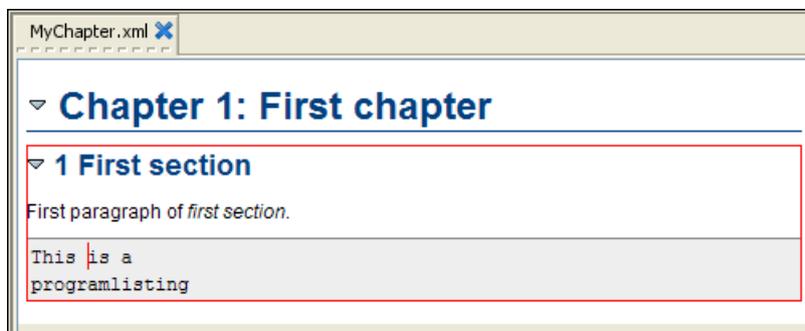
Let's suppose that you want now to insert a second section after the first one. If the `programlisting` is implicitly selected, using Insert After, you'll be able to insert a *subsection* after it, but this is not what you want to do. If you click inside the *para* to implicitly select it and use Insert After, the Edit tool will not suggest *section* at all.

You need to *explicitly select* the first section in order to be able to insert a second section after it. There are many ways to do that.

- Repeatedly press **Ctrl+↑**, while keeping an eye on the node path bar, until you select the desired ancestor element. (Press **Ctrl+↓** if you have gone too far.)
- Repeatedly click inside an element while holding the **Ctrl** key down, without moving the mouse, until you select the desired ancestor element.
- Directly click on the label of the desired element in the node path bar.
- Click on the content generated for the element you want to select: the section number in the case of the `title` of a section, the bullet in the case of a `listitem`, etc.

Note that unlike the implicitly selected element, explicitly selected nodes have a red box drawn around them.

Figure 6. Explicitly selected `section`



Selecting multiple nodes

A number of commands, Convert, Copy, Cut, Paste, Delete, etc, can be applied to multiple sibling³ nodes.

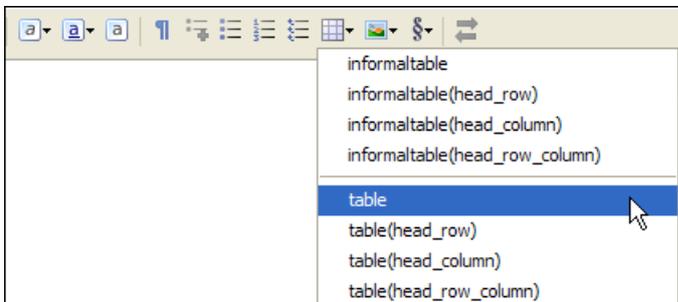
The node selection can be extended to preceding sibling using  Select → Extend Selection to Preceding Sibling (**Esc** ←) and to following sibling using  Select → Extend Selection to Following Sibling (**Esc** →). All the child nodes of an element can be selected using Select → Select All Children (**Esc** ↓).

Alternatively, you can extend the node selection using the mouse by clicking elsewhere (in the direction you want to extend the selection) while holding down the **Ctrl** and Shift keys.

Other commonly used commands

- We have already seen that the DocBook toolbar contains a number of buttons which invoke the Convert command. Most the other buttons (Add para, Add table, Add image, etc) invoke the *add command*.

Figure 7. The "Add table" toolbar button



The add command is very different from the three Insert commands. The add command will find out, where after the caret position, a given element may be inserted. If an insertion position is found, the add command will then insert this given element.

For example, let's suppose that you want to insert an `itemizedlist` after the first `para` of the first section. Click anywhere inside the `para`, *really anywhere, including inside any of its descendant elements* (like its `emphasis` child element), and press the Add `itemizedlist` button of the DocBook toolbar.

- The  Edit → Copy (**Ctrl+C**),  Edit → Cut (**Ctrl+X**),  Edit → Paste (**Ctrl+V**),  Edit → Delete (Backspace, **Del** or **Ctrl+K**) commands work as expected for both the node and the text selection.

Note that in addition to the Paste command (which replaces the explicit selection by the contents of the clipboard or which inserts the contents of the clipboard at caret position when there is no explicit selection), you also have  Edit → Paste Before (**Ctrl+U**) and  Edit → Paste After (**Ctrl+W**) commands.

Also notice that the current content of the clipboard is displayed at the bottom/left of the main window.

Figure 8. The Clipboard tool



- Pressing **Ctrl+Enter** anywhere inside a list item (whatever its element type) or a paragraph allows to quickly add the same element after it.
- Pressing **Enter** anywhere inside a paragraph splits this paragraph in two parts at caret position. This is the most common form of command  Edit → Split (**Esc Enter**).

³Nodes having the same parent element.

- Pressing **Backspace** at the beginning of a paragraph preceded by an element of the same type, merges the two elements. Same behavior if you press **Del** at the end of paragraph followed by an element of the same type. These are the most common forms of command  Edit → Join (**Esc Backspace**).
- Pressing **Ins** allows to quickly add a text node. This is mainly useful in the following situation: you are typing some text in a `para`. You insert a `literal` at caret position and continue typing some text inside the newly inserted `literal`.

Figure 9. Before pressing the Ins key

This para element contains a `literal` child element |

Now you want to resume typing text inside the `para`. You press **Ins** to quickly add a text node after the `literal` and then type text in the newly inserted text node.

Figure 10. After pressing the Ins key

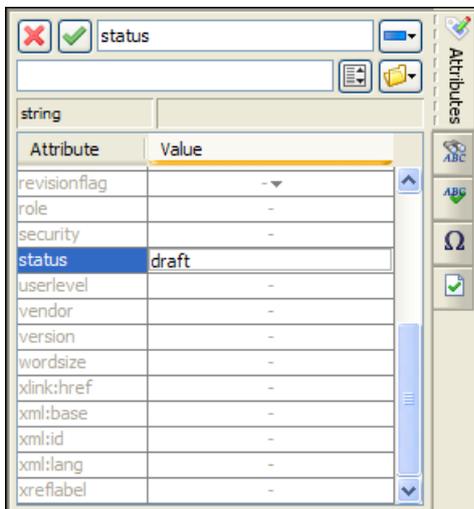
This para element contains a `literal` child element 

- Press **Ctrl+SPACE** to insert a non-breaking space at caret position.

Specifying element attributes

Implicitly or explicitly select an element and use the  Attributes tool to add or modify one or more of its attributes. For example, select the `chapter` root element, click in the cell containing the value of its `status` attribute (found in Value, the right column of the attribute table), type `draft` and then press Enter.

Figure 11. The Attributes tool



However, in most case, you'll want to use the small form found at the top of the attribute table. Example: give an ID to the first `section`:

1. Select the `section`.
2. Press  Tools → Edit Attribute (**Ctrl+E**). This activates the Attributes tool.
3. Type in the first text field (attribute name) the first few letters of the name of the attribute you want to add or modify. Example: type `"xml:id"`.
4. Press **Enter** to select the name and move the keyboard focus to the second text field (attribute value).
5. Type the value of the attribute. Example: type `"introduction"`.

6. Press **Enter** to apply the change and move the keyboard focus to the document view. After that, you can resume the normal editing of your document.

Composing a modular document

Let's suppose you are authoring a `book` but want to edit its `chapter`s separately (because coworkers are working on other `chapter`s of the same `book` or simply because working on small documents is more convenient). How to assemble the separate `chapter`s to form the `book`?

1. Open the document containing the `chapter` in XMLmind XML Editor.
2. Use `File` → `New`, select `DocBook v5+|Book`, then click `OK`. This will create the `book master document`. Now you need to *include* your `chapter document module` in the master document.

3. Switch to the document view containing the `chapter` by clicking on its tab.

4. Select the `chapter` root element and press `■` `Edit` → `Reference` → `Copy as Reference (Ctrl+Shift+C)`.

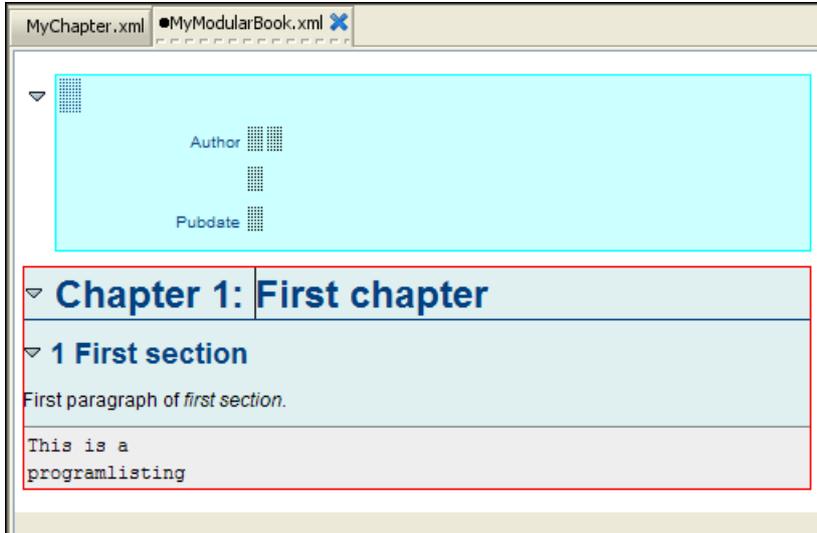
Command `Copy as Reference` allows to copy to the clipboard a *reference* to the root element of a document or to any *element having an ID*.

5. Switch back to the document view containing the `book` by clicking on its tab.

6. Select the first, empty, `chapter` of the newly created `book`.

7. Use `📄` `Edit` → `Paste(Ctrl+V)` to replace this empty `chapter` by a reference to the `chapter` contained in the separate document.

Figure 12. A modular book including a chapter found in a separate document



Notice how the `chapter` included in the `book` is displayed. It has a dimmed blue-gray background, which means that it cannot be edited from within the `book` document⁴.

If you open the modular `book` and want to edit one of its `chapter` modules, click anywhere in an included `chapter` and then use `📄` `Edit` → `Reference` → `Edit Referenced Document (Ctrl+Shift+E)`. This will switch you to the (editable) document view containing the `chapter`. From here, you can switch back to the master document view by using `📄` `Edit` → `Reference` → `Edit Referencing Document (Ctrl+Shift+B)`.

⁴The contents of the included `chapter` cannot be directly modified from within the master document view. However you can freely modify the contents of the `book`. For example, you can remove the `chapter` reference from the `book` or you can cut and paste it elsewhere in the `book`.

2. Menus

2.1. 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 document type (example: DocBook) to which it belongs.

Open

Opens an existing document. Displays standard file chooser dialog box or advanced URL chooser [50] dialog box depending on value of option "Use URL chooser rather than file chooser [56]".

Note that XXE cannot load non-well formed XML documents but it can load invalid documents. See Validity State [47].

Tip

- It is also possible to open a document by dropping its icon anywhere inside XXE main window (except onto "drop sites" such as images in a document view).
- Shift-click on this menu item to open a document in read-only mode.

Save

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

Behaves like Save As if the current document is a newly created document or if the 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 the current document to a different location. Displays standard file chooser dialog box or advanced URL chooser [50] dialog box depending on value of option "Use URL chooser rather than file chooser [56]".

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 the current view of the current document and this, no matter what is being selected. Displays standard Print dialog box.

Print Selected Element

Allows to print a subset of current document. Requires one or more nodes to be explicitly selected (see Select menu [12]). If several nodes are selected, it is the parent element of these nodes which is printed.

Quit

Quit XXE.

Below the above menu items, a menu item is added for the last nine documents that have been opened in 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.

Tip

Shift-click on a menu item to reopen the corresponding document in read-only mode.

2.2. Select menu

2.2.1. 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.

2.2.1.1. 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. • 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.

Using the mouse	Using the keyboard
	<ul style="list-style-type: none"> • Shift+Down or Shift+Up extends the text selection by one line.

2.2.1.2. 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. <p>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.</p> <ul style="list-style-type: none"> • Or Ctrl-click on the node. <p>Ctrl-click several times (without moving the mouse) if needed to. Each Ctrl-click will select the parent of currently selected node.</p> <p>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.</p> <ul style="list-style-type: none"> • 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 [33]. 	<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. <p>Keep an eye on the Node Path bar [33] while doing this.</p> <ul style="list-style-type: none"> • Or use the Select menu [12] or the Select tool bar [35] 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 [33], 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 [33], 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.

2.2.1.3. 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 [33].

All editing commands except the most generic form of Split [16] and Join [17] 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.*

2.2.2. 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**.

2.3. 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 [15], Insert Before [15], Insert [15], Insert After [15], Convert [15], Convert [wrap] [16], Change Processing Instruction Target [18], 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 [63].

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 [14] 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 [35] which can be used to specify an element replacing

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

Insert Before

Displays the Edit tool [35] which can be used to specify an element inserted before

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

Insert

Displays the Edit tool [35] which can be used to specify an element inserted into

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

Insert After

Displays the Edit tool [35] which can be used to specify an element inserted after

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

Convert

Displays the Edit tool [35] which can be used to specify an element replacing

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

Unlike Replace [15] 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</emphasis> girl.</simpara></blockquote>".
```

See also `Wrap` [16] a variant of `Convert`.

Convert [wrap]

This command is a variant of `Convert` [15]. The unique difference between `Wrap` and `Convert` [15] 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` [15]), 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 [35] 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 [16].

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.

2.3.1. 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.

2.3.2. 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.

2.3.3. 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).

2.3.4. Reference menu

About the Include tool [37] feature

Some menu items are by default absent in this menu. You need to enable them by checking "Enable the Include Tool" in Options → Preferences, Features section.

Copy as Reference

Copies to the clipboard a *reference* to the selected nodes (i.e. a pointer to selected nodes). This reference can be later pasted into another document⁵, using any of the normal paste commands — Edit → Paste Before (**Ctrl+U**), Edit → Paste (**Ctrl+V**), Edit → Paste After (**Ctrl+W**) — in places where the grammar constraining the target document allows to do so.

This command is enabled only for documents associated to a configuration declaring a *inclusion scheme* (see Section 12, “inclusionScheme” in *XMLmind XML Editor - Configuration and Deployment*). DITA documents use the `conref` inclusion scheme. DocBook and XHTML documents use the XInclude inclusion scheme.

By default, it is possible to copy as a reference only an element having an ID attribute or the root element of a document. However, for documents using the XInclude inclusion scheme, this restriction may be relaxed by using Options → Preferences, Edit section, Allow advanced use of XInclude [64] checkbox. When this option is turned on, it becomes possible to copy as a reference any range of sibling nodes.

It is not possible to copy as reference the text selection, a reference, descendant nodes of a reference or any node selection directly containing one or more references. If you want to do so, simply use the normal Edit → Copy command, as this command preserves⁶ existing references.

Note

The pasted 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 [20] (or tool bar button ) to open a new window allowing to edit the document containing the referenced nodes.

Replace by Reference

Replace by an element *reference*

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

The element reference to be inserted in the document is specified using the Include tool [37].

Insert Reference Before

Insert an element *reference* before

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

The element reference to be inserted in the document is specified using the Include tool [37].

Insert Reference

Insert an element *reference*

- replacing text selection if any,
- OR, if there is no text selection, insert an element reference in element containing caret, at caret position.

The element reference to be inserted in the document is specified using the Include tool [37].

Insert Reference After

Insert an element *reference* after

- explicitly selected node or node range

⁵It can also be pasted in the same document at another location.

⁶But, unlike Edit → Reference → Copy as Reference, it cannot *create* references.

- OR implicitly selected element.

The element reference to be inserted in the document is specified using the Include tool [37].

Include Text

Displays a file chooser dialog box allowing to choose a text file (of any kind: XML, HTML, .bat, C/C++, etc). The content of this text file is then included in the document being edited at caret position.

This kind of inclusion is implemented by the means of an `<xi:include parse="text">` element. Therefore, this command is disabled unless the document being edited supports the XInclude inclusion scheme.

Untransclude Reference

Replaces included nodes by the inclusion directive (e.g. `xi:include` element).

Any kind of selection inside the included nodes suffices to specify the subject of this command.

Retransclude Reference

Inverse action of Untransclude Reference [20]: replaces inclusion directive (e.g. `xi:include` element) by up-to-date included nodes.

The inclusion directive must be explicitly or implicitly selected.

Using untransclude allows to edit the inclusion directive by hand before retranscluding it. This is useful in the two following cases:

- This allows to add attributes (typically an ID) to the `conref` elements created using Copy As Reference then Paste.
- This allows to fine tune the `xpointer` attribute of the XIncludes created using Copy As Reference then Paste. Example: replace `xpointer="xpointer(id('disclaimer')/*[position() >= 1 and position() <= 8])"` by, simpler and more stable, `xpointer="xpointer(id('disclaimer')/*)"`.

Edit Referencing Document

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

Example 2. Example: book referencing chapters

Book `book.xml` references chapters `chap1.xml`, `chap2.xml`, `chap3.xml`, etc, created in separate documents.

Clicking anywhere inside first chapter displayed in the `book.xml` window then using command Edit → Document Reference → Edit Referenced Document [20] (or tool bar button ) brings the window containing `chap1.xml` to front. (If needed, `chap1.xml` is opened in XXE.)

Now being inside the `chap1.xml` window, using command Edit → Document Reference → Edit Referenced Document [20] (or tool bar button ) brings the window containing `book.xml` to front.

Edit Referenced Document

If the caret or the selection is inside a reference to an element contained in another document, this command brings the window of this other document to front. If the referenced document is not yet opened in XXE, this command will open it.

See example [20] above.

Tip

Shift-click on this menu item to open referenced document in read-only mode.

2.4. Search menu

Search

Displays the Search tool [42] configured for a search session from the current caret position⁷ to the end of the document.

Search Backwards

Displays the Search tool [42] configured for a search session from the current caret position⁷ to the beginning of the document.

Replace

Displays the Search tool [42] configured for a search/replace session from the current caret position⁷ to the end of the document.

Replace Backwards

Displays the Search tool [42] configured for a search/replace session from the current caret position⁷ to the beginning of the document.

Find Next

Search last searched string from the current caret position to the end of the document. The Search tool [42] is not displayed because it is not needed to perform this operation.

Find Previous

Search last searched string from the current caret position to the beginning of the document. The Search tool [42] is not displayed because it is not needed to perform this operation.

Find Element

Displays the Find Element dialog box [52]. 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.

Keyboard shortcut: **Esc f** (**F** like **Find**).

2.4.1. Bookmark menu

Set/Clear

If the caret is located on a bookmark, remove this bookmark. Otherwise, add a bookmark at caret position.

Keyboard shortcut: **Esc b** (**B** like **Bookmark**).

Clear All

Remove all bookmarks, if any.

Go to Last Visited or Go Back

Move the caret to the *last visited bookmark*. If the caret is already located on the last visited bookmark, move the caret back to its initial position. This menu item allows to move back and forth from the place where a user is typing some text to the last visited bookmark.

What is the *last visited bookmark*?

- The last visited bookmark is the bookmark last selected using the menu items described below.
- A newly added bookmark is automatically made the last visited bookmark.

Keyboard shortcut: **Esc j** (**J** like **Jump**).

⁷Turning off the "Start from current caret position" option allows to start searching from the beginning of the document (or from the end of the document depending on the Direction option).

Go to Preceding

Move the caret to the bookmark whose position in the document is before the caret. That makes this bookmark the last visited one (see Go to Last Visited or Go Back [21]).

Go to Following

Move the caret to the bookmark whose position in the document is after the caret. That makes this bookmark the last visited one (see Go to Last Visited or Go Back [21]).

Below the above menu items, a menu item is added for the last nine bookmark which have been added to the document being edited. Selecting a menu item moves to caret to the corresponding bookmark and thus, makes it the last visited bookmark (see Go to Last Visited or Go Back [21]).

2.5. View menu

Redraw

Rebuilds the view of the whole document, no matter which text or nodes are selected.

Updating the references contained in a modular document

This command has also the side-effect of automatically updating *all* the references contained in a modular document (see Copy as Reference [19]). However for a reference to be updated, the referenced document must not have been modified in XXE and not yet saved to disk.

Example: document `book.xml` references `chap1.xml`, `chap2.xml` and `chap3.xml`. File `chap1.xml`, currently opened in XXE, has been modified then saved to disk. File `chap2.xml` is not currently opened in XXE. File `chap3.xml`, currently opened in XXE, has been modified but not yet saved to disk. Using this command will update what has been included from `chap1.xml` and `chap2.xml`, but not from `chap3.xml`, before rebuilding the view of `book.xml`.

Text Size

The items of this menu make it easy changing the base font size of the active styled document view.

Tip

Use your *mouse wheel* while pressing the **Ctrl** key (**Cmd** key on the Mac) rather than use the Larger and Smaller menu items.

Larger

Use a base font size larger by 2pt for the active styled document view. The largest possible base font size is 20pt.

Smaller

Use a base font size smaller by 2pt for the active styled document view. The smallest possible base font size is 10pt.

Normal

Use the "normal", default, base font size (e.g. 12pt). The default base font size is specified in the Preferences dialog box [65].

Use as Default

Apply the current font size to all document views and from now, use it as a default.

This is a handy alternative to specifying the default base font size in the Preferences dialog box [65].

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).

2.6. Tools menu

Declare Namespace

Displays Declare Namespace dialog box [53].

If the current document is conforming to a DTD, the dialog box allows to view the namespaces and their prefixes but not to modify them.

Edit Attribute

Displays Attributes tool [39].

Check Validity

Displays Validity tool [47], 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 the Spell tool [44], unless no spelling errors are found in current document⁸, in which case an OK message is displayed in the status bar.

Automatic Spell Checker

Toggle used to activate and deactivate the automatic (AKA on-the-fly) spell checker. See also option "Activate if this is specified in the configuration file [68]".

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

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 Section 13, "convertCase" in *XMLmind XML Editor - Commands*). This facility is especially useful when recording a macro-command [26].

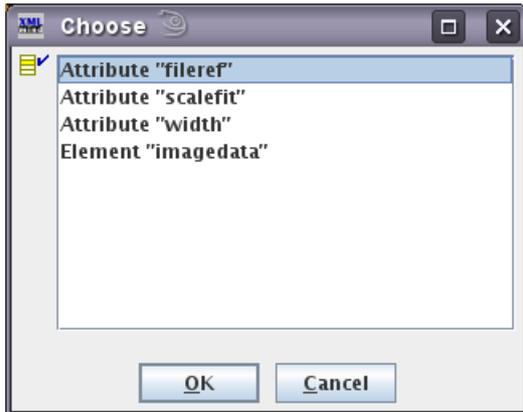
2.6.1. Helper Application menu

This menu contains entries allowing to use third-party applications to view or edit part or all of the document being edited.

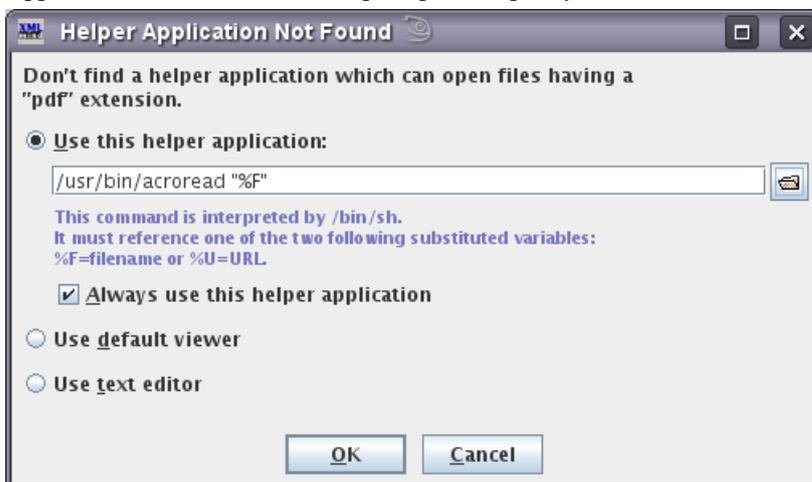
Open Element or Attribute Content in Helper Application

Requires an element to be implicitly or explicitly selected. Displays a dialog box allowing the select this element or one of its attributes from a list.

⁸By default, spell checking starts at the current caret position, but this can be changed by turning off the "Start from current caret position" option of the Spell tool.



Opens specified element or attribute in a third-party "helper" application. If the helper application cannot be determined automatically (because it has not yet been registered using the Preferences dialog box, Helper Applications section), the user is prompted to specify it.



Example of use: use your web browser to open the page referenced in the `url` attribute of the (DocBook 4) `ulink` element.

Edit Element or Attribute Content in Helper Application

Same as above except that the helper application is assumed to be an editor. If this editor is used to modify the element or attribute, then the changes are also automatically applied to the document being edited.

This command works as follows: let's suppose the element of interest contains an image encoded using base 64 (data type `xs:base64Binary`).

1. This command examines the first bytes of the image and, using this signature, determines which helper application to use.
2. If the helper application cannot be determined (because it has not yet been registered using the Preferences dialog box, Helper Applications section), the user is prompted to specify it.
3. It reads the image data from the element, decodes it and saves it to a temporary file.
4. It starts the helper application, an image editor, passing it the file containing the extracted image.
5. After the user quits the image editor, the command detects whether the extracted image has been modified and, if this is the case, reloads it in the element.

Examples of use: use GIMP to edit the image file referenced in the `src` attribute of the (XHTML) `img` element. Use Inkscape to edit the `svg:svg` element contained in an (DocBook 5) `imagedata` element.

Open Document in Helper Application

Opens the document being edited in a third-party "helper" application. If the helper application cannot be determined automatically (because it has not yet been registered using the Preferences dialog box, Helper Applications section), the user is prompted to specify it.

This entry is disabled if the document is newly created and has not yet been saved. Also note that this command is guaranteed to fail if the document is stored on the *Windows file system* and is currently write-locked by XMLmind XML Editor.

Example of use: use your web browser to preview the XHTML document being edited.

Edit Document in Helper Application

Same as above except that the helper application is assumed to be an editor. XMLmind XML Editor will automatically reload the document after the third-party editor is used to modify it.

This entry is disabled if the document is newly created and has not yet been saved. Also note that this command is guaranteed to fail if the document is currently write-locked by XMLmind XML Editor.

Example of use: use your favorite text editor to perform some low-level modifications on the XML document being edited.

2.6.2. Spreadsheet menu

 This menu is available only in XMLmind XML Editor Professional Edition, where it is hidden by default. You need to enable it by checking "Enable the Integrated Spreadsheet Engine" in Options → Preferences, Features section.

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 [68].

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 [68].

Update

Forces a full calculation of the spreadsheet.

2.6.3. "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 [12] facility or with the Search [42] 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 Section 2, "command" in *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 [35], Attributes [39], Search [42], Characters [46].

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 View → Add 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 [13], Redo [13], Repeat [13],
- 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 [15] 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 [23] 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.

2.7. 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").

2.8. Options menu

Preferences

Displays the Preferences dialog box [56] which allows to specify a large number of user preferences.

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 (i.e. .jar files).

 This menu item is available in all editions of XMLmind XML Editor, but it is hidden by default. You need to enable it by checking "Enable the Developer Tools" in Options → Preferences, Features section.

Install Add-ons

Displays the Install Add-ons dialog box [77], which allows to download and install, upgrade and uninstall all sorts of add-ons. This dialog box is not available if XXE has been deployed using Java™ Web Start.

Tip

If you hold the Shift key and click on Options → Install Add-ons, you'll be able to *automatically* upgrade all the installed add-ons (of course, if any and if needed to).

Note

This menu item is absent when XMLmind XML Editor has been started using Java™ Web Start.

2.8.1. "Quick Preferences" menu

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.

Default: checked.

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

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.

Default: checked.

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

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.

Default: checked.

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

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.

Default: checked.

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

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.

Default: not checked.

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

2.8.2. "Customize Configuration" menu

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

This menu allows end-users (i.e. non-experts) to customize the configuration associated to the document being edited.

End-user customization of configurations by the means of this menu also works when XMLmind XML Editor is deployed using Java™ Web Start.

The following menu items have an effect on a *single* configuration: the configuration of the document being edited. For example, if you specify DocBook XSL style sheet parameter `paper.type=USLetter` when you are editing a DocBook document, this will have no effect on the DocBook 5 and Simplified DocBook configurations, even if these configurations share their Convert Document menu with the DocBook configuration.

Using any of the following menu items has an immediate effect on all the opened documents associated to the customized configuration. That is, no need to restart XXE to benefit from the customization.

Save Document As Template

Save the document being edited as a named template. This named template is then displayed in the File → New dialog box.

This menu item is disabled (grayed) if the document being edited has been modified and thus, needs to be saved. Therefore you need to save the document being edited to disk prior to using this facility.

Arbitrarily complex documents may be used as templates: a document may be modular, may reference graphics files, etc. This is needed for example to create a document template including the logo of a company.

User-defined document templates are added and removed using the "Save Document As Template" dialog box [80].

Save Selected Element As Template

Save the selected element as a named template. This named template may then be suggested by the Edit tool, depending on the operation to be performed and if this is allowed by the schema.

This menu item is disabled (grayed) unless an element is explicitly selected.

User-defined elements templates are added and removed using the "Save Selected Element As Template" dialog box [81].

Save Views As Default

Save the current layout of views as the default one. Displays a simple confirmation dialog box.

Example 1: you want the CSS style sheet called "Images displayed as thumbnails" to be used by default for all your DocBook documents.

1. Open a DocBook document.
2. Select CSS style sheet called "Images displayed as thumbnails" in the View menu.
3. Use Options → Customize Configuration → Save Views As Default.
4. Open another DocBook document (or use File → New to create a new one) to check that your customization has worked.

Example 2: when you open a DocBook document, you want to see the styled view in the middle, the tree view at the left and the structure view (i.e. CSS style sheet called "Document Structure") at the bottom.

1. Open a DocBook document.

2. Use View → Add. Select "(no style sheet)" for the Left pane. Select "Document structure" for the Bottom pane.
3. Use Options → Customize Configuration → Save Views As Default.
4. Open another DocBook document (or use File → New to create a new one) to check that your customization has worked.

Add Keyboard Shortcut

Add keyboard shortcuts. In other terms, bind up to two keystrokes (example: "F2 u" to a command (example: "moveElement up[implicitElement]").

User-defined keyboard shortcuts are added and removed using the "Add Keyboard Shortcut" dialog box [84].

Change Document Conversion Parameters

Change the XSL style sheet parameters (e.g. `paper.type=USLetter`) used when converting the document being edited to other formats (e.g. PDF, HTML, etc).

User-defined XSL style sheet parameters are specified using the "Change Document Conversion Parameters" dialog box [88].

2.9. 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.

2.10. Help menu

Getting Started

Displays a short tutorial in the help browser.

Help

Displays the help browser.

Help About

In order to display the help section related to a specific GUI component of XXE, first 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.

✚ This menu item is available in all editions of XMLmind XML Editor, but it is hidden by default. You need to enable it by checking "Enable the Developer Tools" in Options → Preferences, Features section.

Plug-ins

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

✚ This menu item is available in all editions of XMLmind XML Editor, but it is hidden by default. You need to enable it by checking "Enable the Developer Tools" in Options → Preferences, Features section.

Install Evaluation Key

Allows to transform XMLmind XML Editor Personal Edition to an *Evaluation Edition* by installing an *evaluation key*.

XMLmind XML Editor Evaluation Edition has exactly the same features as Professional Edition. It may not be used for purposes other than the evaluation of the Professional Edition product. The evaluation period lasts seven days. After that, XMLmind XML Editor automatically and silently returns to its initial Personal Edition state.

This menu item displays a dialog box allowing to specify the file containing the evaluation key.

This evaluation key must have been downloaded from <http://www.xmlmind.com/xmlmind/evaluate.html>.

This menu item basically checks the validity of the evaluation key and then copies it to `XXE_user_preferences_directory/evaluationKey.properties`. After that, suffice to restart XMLmind XML Editor to activate all the features of Professional Edition.

✚ This menu item is *absent* from XMLmind XML Editor Professional Edition.

About XMLmind XML Editor

Displays the customary About dialog box.

3. The document view area

Figure 13. Tab corresponding to the active document

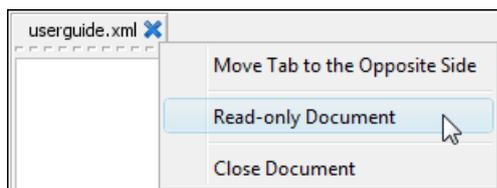
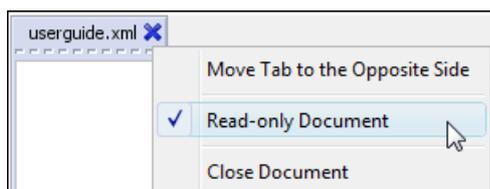


Figure 14. Tab corresponding to the active document when it is in read-only mode



Move Tab to the Opposite Side

Moves this tab to the opposite side of the document view area. When several documents are opened in XMLmind XML Editor, this has the effect to split the document view area in two parts. See also Options → Quick Preferences → Split Windows Vertically [16].

This is equivalent to clicking on the dashed line found at the bottom of a tab.

Read-only Document

This check box may be used to switch the active document from its normal, read-write, mode to a read-only mode and the opposite way round.

Making a document read-only is useful in the following cases:

- You want to open a document without locking it (because doing so would prevent your coworkers from modifying it).
- You want to make sure that you'll not be able to modify the opened document by mistake.

Note that there are several ways to directly open a document in read-only mode. Example: Shift-click on File → Open [9] menu item or on the Open button of the main tool bar [32].

Close Document

Closes the active document and switches to the next document opened in XMLmind XML Editor, if any.

This is equivalent to clicking on the blue cross found at the right of the tab.

4. Main tool bar



New

See File → New [9].



Open

See File → Open [9].

Tip

Shift-click on this button to open a document in read-only mode.



Save

See File → Save [9].



Save All

See File → Save All [9].



Use URL chooser rather than file chooser

See option Use URL chooser rather than file chooser [56].



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



Undo

See Edit → Undo [13].



Redo

See Edit → Redo [13].



Repeat

See Edit → Repeat [13].

 Command History
See Edit → Command History [13].

 Cut
See Edit → Cut [13].

 Copy
See Edit → Copy [13].

 Paste Before
See Edit → Paste Before [14].

 Paste
See Edit → Paste [14].

 Paste After
See Edit → Paste After [14].

 Delete
See Edit → Delete [14].

 Split
See Edit → Split [16].

 Join
See Edit → Join [17].

5. Node path/node selection bar

5.1. Document reference tool bar

 Edit Referencing Document
See Edit → Document Reference → Edit Referencing Document [20].

 Edit Referenced Document
See Edit → Document Reference → Edit Referenced Document [20].

Tip

Shift-click on this button to open referenced document in read-only mode.

5.2. 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.
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).
Drag the file icon 	Drag and drop in another application, the <i>location</i> of the document being edited in XXE. Do not forget to save the document being edited in XXE before dropping its location in a viewer application.

5.2.1. 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 [19].



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 [33] which are at the left of the node path bar.

5.2.2. 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*.

5.3. Select tool bar

Select Parent

See Select → Select Parent [12].

Select Child

See Select → Select Child [12].

Select Preceding Sibling

See Select → Select Preceding Sibling [12].

Select Following Sibling

See Select → Select Following Sibling [12].

Find Element

See Search → Find Element [21].

6. 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.

6.1. Edit tool

This tool works in two steps:

1. Click on one of the following buttons:

Replace

(See Edit → Replace [15] for a complete description of the command.)

Insert Before

(See Edit → Insert Before [15] for a complete description of the command.)

 **Insert**

(See Edit → Insert [15] for a complete description of the command.)

 **Insert After**

(See Edit → Insert After [15] for a complete description of the command.)

 **Convert**

(See Edit → Convert [15] for a complete description of the command.)

 **Convert [wrap]**

(See Edit → Convert [wrap] [16] 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 [63].

- 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 [23] 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 [23] 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.

6.2. Include tool

Note

The Include tool is hidden by default. You need to enable it by checking "Enable the Include Tool" in Options → Preferences, Features section.

6.2.1. Working with ``XML variables``

The Include tool is designed to make it quick and easy inserting ``XML variables`` in your documents.

XML variables are often-used contents: product names, product versions, copyright information, addresses, phone numbers, etc, you need to reference in almost all your documents.

XML variables are best modelled by elements having a weak semantic and which may be inserted almost anywhere in a document. In the case of XHTML, this element is `span`. In the case of DocBook, this element is `phrase`.

XML variables are generally collected in a single, centralized, ``pseudo-document``, created for this sole purpose.

You'll insert *references* to XML variables, and *not copies*, in your actual documents.

By working this way, if one day, the value of an ``XML variable`` changes, you don't need to manually update all the documents making use of this value.

Unlike the Copy As Reference/Paste (**Ctrl+Shift+C/Ctrl+V**) approach described in the tutorial, which is a generic way to compose any kind of modular documents, the Include tool is specialized in XML variables. For example, do not attempt to use this tool to insert in a modular book the references to its chapters.

6.2.2. Using the Include tool

First of all, specify the filename or URL of the document containing all your XML variables.

This is done once for all. Up to 5 documents may be added to the list displayed by the "Contained in document" combobox. Specified documents are remembered in order to be reused in subsequent editing sessions.

- Use the  button to add a filename or URL to the list displayed by the "Contained in document" combobox.
- Use the "Contained in document" combobox to select the document containing your XML variables.
- Use the  button to remove currently selected document from this list.

If you attempt to add an invalid (non-XML or non well-formed) or useless (contains no elements having an ID) document to the list, you'll be informed of this fact just once and the document will nevertheless be added to the list. In such case you'll have to use the  button to explicitly remove it from the list.

After that, the tool works in two steps:

1. Click on one of the following buttons:

-  Replace by Reference
(See Edit → Reference → Replace by Reference [19] for a complete description of the command.)
-  Insert Reference Before
(See Edit → Reference → Insert Reference Before [19] for a complete description of the command.)
-  Insert Reference
(See Edit → Reference → Insert Reference [19] for a complete description of the command.)
-  Insert Reference After
(See Edit → Reference → Insert Reference After [19] for a complete description of the command.)

This first action just fills the list below these buttons with all the allowed arguments (element ``identifiers" and/or "-") for the selected command.

This list will only show elements:

- directly contained (that is, not themselves included) in the document selected by the "Contained in document" combobox,
- AND which may be copied as reference (generally this means that such element has an ID),
- AND allowed by the schema given the chosen command and the current selection.

The root element, when listed, is specified as "-".

2. Choose the identifier of the element (or "-") from the list or type it in the text field. It is this second action which actually triggers the command.

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

- The identifier of the element (or "-") is selected by clicking on it in the list. A single click is sufficient.
- Alternatively, the identifier of the element (or "-") can be specified using the text field above the list:

1. Type the identifier of the element (or "-") in the field.
2. Press Enter or click on the  OK button.

Tip

This text field supports auto-completion.

The auto-completion feature can be configured using the Options dialog box [63].

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

6.3. 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.

6.3.1. 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:

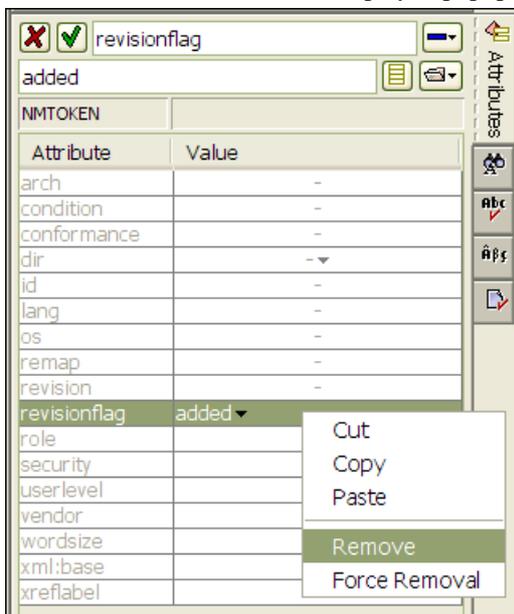
1. Click on the field at the right of the attribute name. This field is editable.
2. Type the value of the attribute.

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.



6.3.2. 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 [63].

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 [23] 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 [23] 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 Preferences dialog box [56]).

6.4. 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".

Tip

When "Ignore case" has been checked, "Regular expression" has not been checked and both the searched string and the replacement string contain only lower-case letters or the hyphen character ('-'), the replacement string actually used is capitalized exactly like the found string it will replace.

Example: if you specify "editor" as the search string and "modeller" as the replacement string, the Search tool will replace "editor" by "modeller", "Editor" by "Modeller" and "EDITOR" by "MOD-ELLER".

Counterexample: if you specify "IBM" as the search string and "Lenovo" as the replacement string, the Search tool will replace "IBM", "Ibm" and "ibm" by "Lenovo".

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

Search from the current caret position to the beginning of the document.

Down

Search from the current caret position to the end of the document.

Wrap search

Restart the search from the beginning of the document when the end of the document has been reached. (If Direction is Down, restart the search from the end of the document when the beginning of the document has been reached.)

6.5. Spell tool

Procedure for checking current document for spelling errors:

1. Click on the Start button or use Tools → Check Spelling [44].
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.

Default language

Selects the language of the dictionary used by the spell checker *when such language is not specified in the element being checked for spelling* (typically through the use of an attribute such as `lang` or `xml:lang`).

It is possible to switch from a default language to another at any time. In such case, when the default language is actually used (that is, when `lang` or `xml:lang` attributes have not been specified), 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 default language is recorded in the user preference file in order to be automatically chosen in subsequent XXE sessions.

Example 3. Default language versus actual language example

In this example,

- `Test.html` does not have `xml:lang` attributes set on its elements, which means that the default language, English, is used to spell check this document.
- Except, in `span` containing "le petit cha est mort" where attribute `xml:lang` has been set to `fr`. This means that for that element only, the default language, English, is ignored and that a French dictionary will be used.



In the above screen shot, notice that the actual language of `span` "le petit cha est mort", `fr`, is displayed at the top/right of the Spell tool.

Start from caret

If this option is turned on, spell checking starts from the current caret position. If this option is turned off, spell checking starts from the beginning of the document.

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".

About learned words

When a spell checking session is started, it automatically knows all the words learned during past and current spell checking sessions.

A ``spell checking session" is started each time:

- You click the Start button of the Spell tool.
- You open a document and auto spell checking is turned on for this document.
- Auto spell checking was turned off and you manually turn it on for the document being edited. This is done for example, by clicking the "Automatic Spell Checker" button found at the left of the status bar.

Limitations:

- Words learned during past and current spell checking sessions are acquired when a new session is started, and never in the middle of a session.

Example: let's suppose auto spell checking is turned on for all documents. Document A and document B are both opened in XXE. You use the popup menu of the automatic spell checker to force XXE to learn word "foo" in document A. You switch to document B which also contains word "foo". Word "foo" is still underlined in document B.

- "Ignore All" words and "Replace All" words are not shared by spell checking sessions.

6.6. 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").

6.6.1. 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.

6.7. 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 [23] or Validity State [47],
- OR File → Save [9] or File → Save As [9] (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 [47].

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.

7. Status bar

7.1. Automatic Spell Checker

 Automatic Spell Checker

See Tools → Automatic Spell Checker [23].

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

7.2. Validity state

Clicking on this button displays Validity tool [47], 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>validateHook</code> reports such warnings (See Section 29, “ <code>validateHook</code> ” in <i>XMLmind XML Editor - Configuration and Deployment</i>).

Icon	Meaning
	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>validateHook</code> reports such errors (See Section 29, “ <code>validateHook</code> ” in <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.
	This document has severe validity errors such 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> Tip After fixing some structure errors using XXE, save and then reload the document. 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.

7.3. 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.

7.4. Overwrite mode

INS/OVR

INS indicates *Insert Mode*: typing a character inserts it at caret position.

OVR indicates *Overwrite Mode*: typing a character *replaces* the character found at caret position by the typed character. If the caret is positioned at the very end of a text (or comment or processing-instruction) node, then typed characters are simply inserted there.

Pressing the INS/OVR button allows to switch from Insert Mode to Overwrite Mode and vice versa. The shortcut associated to this button is by default **Esc Insert** (press **Esc**, release **Esc**, then press **Insert**).

7.5. 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 [14], Edit → Paste [14], Edit → Paste After [14] 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.

7.5.1. 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 [14] or type **Ctrl+V**.

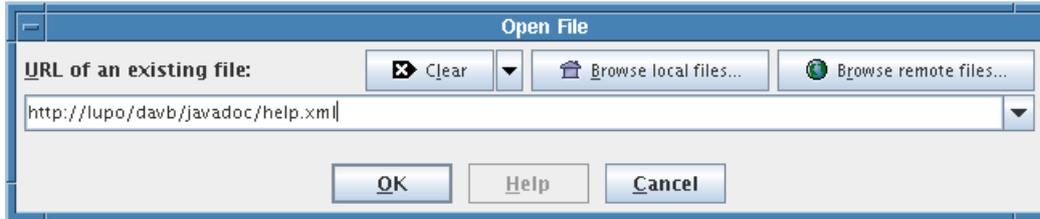
8. Dialog boxes

8.1. The URL chooser dialog box

The URL chooser dialog box allows to specify the location of a file or directory on the local file system and, if *virtual drive plug-ins have been installed*⁹, on remote (e.g. HTTP, FTP) file systems. 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.

 This dialog box is available only in XMLmind XML Editor Professional Edition.

This dialog box is similar to the "Open Web Location" dialog box found in many Web browsers. To use it, simply type in its text field the URL (Uniform Resource Locator — see syntax [52] below) of a file or directory.



Now rather than typing an URL, you can *select it* using a file chooser dialog box:

Browse local files

Allows to choose a local file or directory using the standard file chooser dialog box.

Always works, whatever is specified in the text field.

Browse remote files

Allows to choose a file or directory found on the local file system and, if *virtual drive plug-ins have been installed*, on remote (e.g. HTTP, FTP) file systems.

Cannot work if the text field is totally blank. You have to type¹⁰ at least the URL of the initial directory you would like to browse.

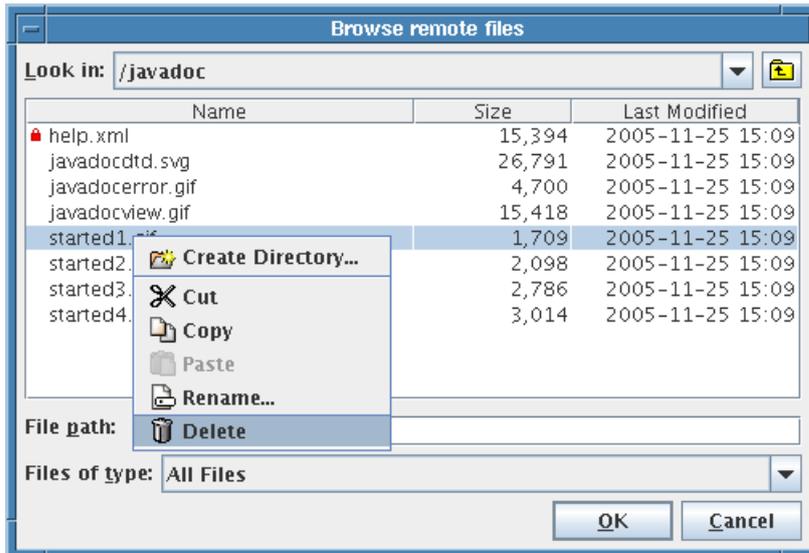
Note that the special chooser dialog box displayed after clicking on this button has a simple yet handy, integrated file manager. To use it, simply right-click anywhere on the file list:

⁹A virtual drive plug-in allows XMLmind XML Editor to edit documents stored in places other than the local file system. This is done by emulating a hierarchical file system.

Installed plug-ins can be listed using Help → Plug-ins [31].

All sorts of add-ons, including virtual drive plug-ins, can be downloaded and installed using Options → Install Add-ons [28].

¹⁰Or choose an item from the dropdown menu next to the text field. This menu lists last recently visited URLs.



Clear

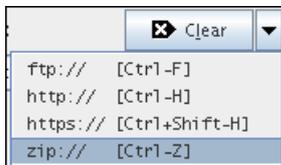
Erases the contents of the text field where the URL is typed.

The dropdown menu next to the Clear button

(Disabled unless virtual drive plug-ins have been installed.)

Instead of erasing the text field, this menu allows to replace its contents by the protocol part of an URL, e.g. "http://", "ftp://", etc. The list of these prefixes depend on which virtual drive plug-ins have been installed.

Moreover, keyboard shortcuts are automatically associated with the prefixes. For example, press **Ctrl+H** to replace the contents of the text field by "http://".



Specifying the "http://" URL of a document found on a WebDAV server

Assumes that WebDAV virtual drive plug-in has been installed and that the option "Use URL chooser rather than file chooser" has been selected, that is, you see  in the tool bar.

Before using the Browse remote files button, make sure to type in the text field the URL of an initial directory *which is WebDAV enabled*.

Example: you have typed "http://www.acme.com/" as an initial URL. After clicking on the Browse remote files button, you get the following error message: "Cannot connect to http://www.acme.com/: http://www.acme.com/ does not support WebDAV". Try to type the URL of a subdirectory such as "http://www.acme.com/publish/" because, may be in the "www.acme.com" server, directory "/" is not WebDAV enabled, while directory "/publish/" is.

Syntax of Uniform Resource Locators

An *URL* (Uniform Resource Locator), often called *Internet Address* in Web browsers, has (to make it simple) three parts:

1. The protocol part. Example: "http://" in URL "http://www.xmlmind.com/xmleditor/download.shtml".

Unlike Web browsers, XMLmind XML Editor requires you to specify the protocol.

2. The authority part, often a server, sometimes empty. Example: "www.xmlmind.com" in URL "http://www.xmlmind.com/xmleditor/download.shtml".

Example of URL with an empty authority: `file:///C:/temp/documentation.zip`. Because the authority is empty, this URL could be specified as `file:/C:/temp/documentation.zip`.

3. The file path part. Example: "/xmleditor/download.shtml" in URL "http://www.xmlmind.com/xmleditor/download.shtml".

File path components, that is, directories, are separated by forward slashes ('/'), whatever is the platform running XMLmind XML Editor.

The file path of a directory must end with a '/'. Example: "http://www.xmlmind.com/xmleditor/" is the URL of directory `/xmleditor/`.

More examples:

- `http://www.xmlmind.com/store/buy.php`
- `http://lupo.pixware.fr/~john/`
- `https://www.acme.com:8080/registry/customers.xml` (8080 is the port number of the `www.acme.com` Web server)
- `ftp://zenda.prisoner.tv/pub/docs/refman.zip`
- `ftp://john:sesame@zenda.prisoner.tv/pub/docs/refman.zip` (same as above with a username "john" and password "sesame").
- `file:///home/john/documentation.zip`
- `file:/home/john/`
- `file:///C:/Documents%20and%20Settings/john/documentation.zip` (spaces must be escaped as "%20")

8.2. The "Find Element" dialog box

This dialog box allows to select nodes specified using an XPath expression. This dialog box has a Simple tab [52] which allows to perform most common search tasks without having to learn XPath. Arbitrarily complex XPath expressions are specified using the Advanced tab [53].

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

8.2.1. 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.

8.2.2. 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.

8.3. The "Declare Namespace" dialog box

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 an xmlns attribute to elements to specify an namespace and its prefix. 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.

However, when the current document is conforming to a DTD, this dialog box can only be used to view the namespaces and their prefixes and not to edit them.

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.

8.4. The spreadsheet 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.

8.5. The "Preferences" dialog box

8.5.1. New options

Immediately save newly created document

When a new document is created using File → New, File → Open as Template or File → Open Copy and this toggle is checked, XXE will immediately prompt the user for a save file name. This save file name is used to actually create on disk the file corresponding to the new document.

Note that even when this toggle is *not* checked, XXE may immediately prompt the user for a save file name. This will happen for *composite document templates*, that is, document templates referencing graphics files and/or including document modules.

Default: not checked.

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

8.5.2. Open options

Use URL chooser rather than file chooser

If this toggle is checked, XXE will display an URL chooser dialog box [50] rather than the standard file chooser dialog box each time the user needs to specify a file name.

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

Default: not checked.

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

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 DTD

If this toggle is checked, a dynamic, non constraining, pseudo DTD is used for documents not conforming to a DTD or schema.

The pseudo DTD simply remembers all attributes and child elements added to each element during the editing session. Thanks to this pseudo DTD, 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`.

Inform about non-editable document parts

If this toggle is checked and if the document being opened is modular (that is, contains references to nodes coming from other documents), XXE display a dialog box informing the user that included nodes have been marked as non-editable.

Default: checked.

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

Lock documents stored on the local filesystem

If this toggle is checked, the file containing the document being opened in the editor is automatically locked using an *advisory, application-level, lock*.

In practice, this means that:

- Applications other XXE can open and modify the document being edited in an instance of XXE.
- If you try to open in XXE a document being edited in another instance of XXE (run by you or by one of your coworkers), XXE will inform you that the file is locked and will tell you who is locking it. After that you'll be able to open the document but, if you modify it, you'll have to save it to another location.

Caveat: In order to use this facility, you and *all* your coworkers possibly editing the same files, need to turn this option on. If one of your coworkers forgets to do it, XXE will not be able to detect that the file being opened is locked, and in such case, your coworker may overwrite a document being edited by another person.

How to forcibly remove a lock

A lock is implemented by the means of a hidden text file containing information about who has locked the corresponding document.

On Windows, the lock file corresponding to document `C:\doc\doc.xml` is `C:\doc\doc.xml.LOCK` (with the hidden attribute set on this file).

On the other platforms, the lock file corresponding to document `/home/john/doc/doc.xml` is `/home/john/doc/.doc.xml.LOCK`.

Manually deleting the lock file forcibly removes the lock set on the corresponding document. Note that it is harmless to delete such lock files and this, at any time.

Default: not checked.

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

When possible, lock documents stored on remote filesystems

Unlike the above toggle, this form applies only to files stored on filesystems other than the local one (WebDAV, FTP, Zip, etc).

 This group of options is present only in XMLmind XML Editor Professional Edition.

Preferred lock mode

Specifies the type of lock [58] used for the document being edited, *when the underlying document store supports locking*. Currently, only WebDAV supports document locking.

Note that the mode specified here is just a wish. XXE will adapt the lock mode to what is actually supported by the underlying document store. For example, if the Shared mode has been chosen by the user and the underlying document store only supports Exclusive locking, the document being edited will be locked using an Exclusive lock.

Default: Exclusive

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

Identifier as a lock owner

Specifies a free form string identifying you as the person locking a document. This string is presented to other users wishing to know who is locking a document. It is recommended to specify your email address (example: jdoe@acme.com).

Default: empty string which implies *user_name@host_name* (example: jdoe@fast.acme.com)

Preference key: lockOwner; type: string; default: empty string which implies *user_name@host_name*.

Table 2. 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.

8.5.3. 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: the original encoding of the file.

Preference key: encoding; type: any encoding supported by Java or "ORIGINAL_ENCODING"; 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: `nbsp 160 0xA0 0240 reg 174 0x00ae 0256 pound:yen 163:165 0xA3:0xA5 0243:0245`

Default: empty.

Ignored if toggle "Save characters outside encoding as entity references [58]" is not checked.

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 3. XHTML examples

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

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 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.

Ensure that a save filename has an extension

If this toggle is checked, XXE automatically appends an extension to the save filenames (File → Save As, File → Save Copy, etc) having no extension at all.

The extension is guessed based on the current filename of the document to be saved. For example, if you save `/tmp/page.html` as `/home/john/index`, XXE automatically appends `".html"`. When the extension cannot be guessed this way, XXE fallbacks to `".xml"`.

Default: checked.

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

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`.

Automatically update references in modular documents

If this toggle is checked, references found in modular documents are automatically updated each time a referenced document is saved to disk.

Example: document `book.xml` references `chapter1.xml` and `chapter2.xml`. File `chapter2.xml` is modified and saved to disk using XXE. Nodes included in `book.xml` coming from `chapter2.xml` are automatically updated.

Counter-example: document `book.xml` references `chapter1.xml` and `chapter2.xml`. File `chapter2.xml` references `section1.xml` and `section2.xml`. File `section2.xml` is modified then saved to disk using XXE. Nodes included in `book.xml` *indirectly* coming from `section2.xml` are *not* automatically updated. In such case, if you really want to update `book.xml`, you'll have to use View+Redraw (**Ctrl+L**) See note about Updating the references contained in a modular document [22].

Default: checked.

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

8.5.4. 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 [62]

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 View options [65] 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 4. Substituted Variables

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

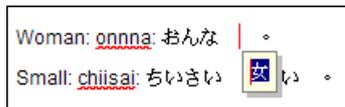
8.5.5. Edit options

Use integrated input method support

When an input method (e.g. Japanese Romaji Input) has been selected by the user,

- If this toggle is checked, a composition area, managed by the XML editor, is displayed inside the document view, below the caret.
- If this toggle is not checked, a composition window, managed by the operating system, is displayed outside the main window of the XML editor.

Figure 15. Composition area displayed below the caret



Default: not checked. (Slightly more efficient if you never use input methods.)

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

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 margin to the styled view

If one or both of these toggles are checked, the editor adds gray margins at the left and/or at the right of the styled document views. These margins allow 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: `addInteractiveMargin`; type: `none`, `left`, `right` or `both`; default: `none`.

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.

Automatically select an attribute

When toggle Never is not checked, Tools → Edit Attribute (**Ctrl+E**) automatically selects an attribute of the element. The selected attribute is determined as follows, in this order:

1. First required attribute containing the "???" placeholder value.
2. First attribute, required or not, containing the "???" placeholder value.
3. If toggle Unique required attribute is checked, *unique* required attribute, whatever its value.

Default: First attribute containing a placeholder value.

Preference key: `autoSelectAttribute` ; type: never, placeholder or required; default: placeholder.

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.

Allow advanced use of XInclude

By default, command Edit → Reference → Copy as Reference [19] allows to copy as a reference only the root element of a document or an element having an ID. If this toggle is checked, it becomes possible to copy as a reference any range of sibling nodes.

Default: not checked.

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

Warn about advanced use of XInclude

This checkbox is disabled unless "Allow advanced use of XInclude [64]" has been checked.

If this toggle is checked, you'll be informed when what you are copying as a reference qualifies as "an advanced use of XInclude". Typically you'll be informed that the element you are copying as a reference has no ID attribute.

If you check Allow advanced use of XInclude [64], it is also strongly recommended to also turn this option on.

Default: not checked.

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

8.5.6. 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`. See also View → Text Size [22].

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.)

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`.

Activation of the automatic (AKA on-the-fly) spell checker.

 This group of options is present only in XMLmind XML Editor Professional Edition.

Never activate the automatic spell checker

When a document is opened, never activate the automatic spell checker, even if this is specified in the configuration file associated to the opened document.

Default: not checked.

Always activate the automatic spell checker

When a document is opened, always activate the automatic spell checker, even if this is *not* specified in the configuration file associated to the opened document.

Default: not checked.

Activate if this is specified in the configuration file

When a document is opened, activate the automatic spell checker if this is specified in the configuration file associated to the opened document (using the `spellCheckOptions` configuration element, see Section 23, "spellCheckOptions" in *XMLmind XML Editor - Configuration and Deployment*).

Default: checked.

Preference key: `autoSpellCheck`; type: `enumeration` (`never`, `always`, `seeConfig`); default: `seeConfig`.

8.5.7.2. Validate options

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`.

Filter duplicate ID errors found in modular documents

Pasting in a document two references to the same element will cause the Validity tool to display duplicate ID errors. This is a problem because the author didn't do any mistake by pasting these two references and because too many of these ``false" duplicate ID errors may hide real duplicate ID errors. That's why turning on this option will discard such ``false" duplicate ID errors from the Validity tool.

Default: checked.

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

8.5.7.3. Spreadsheet options

 This group of options is available only in XMLmind XML Editor Professional Edition, where it is hidden by default. You need to enable it by checking "Enable the Integrated Spreadsheet Engine" in Options → Preferences, Features section.

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 [26].

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 [25] and also disabling the auto-update mode [26] are recommended.

Default: checked.

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

8.5.7.4. Helper applications options

File types

List of file types. Each file type has an associated helper application. This helper application is assumed to be able to open files detected as having this type. A helper application may be a viewer or an editor.

Default: the `"text/plain"` file type:

- On Windows: `text/plain:txt:::notepad "%F"`
- On the Mac:
 - for a version older than Mac OS X 10.5 (e.g. Tiger): `open -t "%F"`,
 - starting from Mac OS X 10.5 (Leopard): `open -W -n -t "%F"`.

See also [Helper applications on the Mac](#) [80].

- On Unix: `text/plain:txt:::xterm -e vi "%F"`

Buttons acting on this list:

Add

Displays the "Helper Application Editor" dialog box [78] in order to add a new file type to the list.

Edit

Displays the "Helper Application Editor" dialog box [78] in order to view or modify selected file type.

Remove

Removes selected file type from the list.

Preference key: `helperApplications.entries`; type: string; default: see above. The format of this string is:

```
entries --> ( entry ('\n' entry)* )?
entry --> mime_types ':' extensions ':' magic_strings ':'
        xml_name_patterns ':' command

mime_types --> (mime_type (',' mime_type)* )?
extensions --> (extension (',' extension)* )?
magic_strings --> (magic_string (',' magic_string)* )?
xml_name_patterns --> (xml_name_pattern (',' xml_name_pattern)* )?

In any of the above fields, character ':' must be escaped using "\072"
and character ',' must be escaped using "\054".

magic_string --> HEXADECIMAL_NUMBER

xml_name_pattern --> ( '{' namespace_URI? '}' )? local_part
One of local_part or namespace_URI may be equal to "*".
```

Default viewer

Specifies which default viewer to use in case the type of the file to be opened has not been detected. In practice, commands making use of the default viewer typically assumes that it is in fact *a Web browser*. This implies that these commands assume that a default viewer can open URLs as well as filenames and that it can open text, html, GIF, PNG and JPEG files.

This field must contain a command line interpreted by the native shell of the platform: `cmd.exe` on Windows and `/bin/sh` on the Mac and on Unix.

This command line must reference one of these two substituted variables: `%U` and `%F`. In principle, `%U` is replaced by the URL of the file to be opened by the helper application and `%F` is replaced by a filename. In practice, `%U` is just a *hint* meaning: the helper application can open URLs as well as filenames.

Default: depends on the platform:

- On Windows: `start "" "%U"`
- On the Mac: `open "%U"`
- On Unix: dynamically detected. By default: `(mozilla -remote "openURL(%U)" 1> /dev/null 2>&1) | (mozilla "%U" &)`

Buttons acting on this field:

Reset

Resets the field to its default value (see above).

Choose

Displays the standard file chooser in order to specify an application (e.g. a `.exe` or a `.bat` file on Windows). String `" %F"` is automatically appended to the chosen application.

See also Helper applications on the Mac [80].

Preference key: `helperApplications.defaultViewer`; type: string; default: see above.

8.5.8. 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.

8.5.9. Install add-ons options

Note

This preference sheet is absent when XMLmind XML Editor has been started using Java™ Web Start.

Download add-ons from these servers

Add-ons available for download are listed in `.xxe_addon` files. The list of ``servers" below this checkbox contains the URLs of these `.xxe_addon` files. Use button Add to add an URL to the list. Use button Remove to remove selected URL from the list. Use button Reset to reset the list to its default value.

Default: checked. The default value of the list of ``servers" is:

- http://www.xmlmind.com/xmleditor/_usercontrib/list.xxe_addon
- [http://www.xmlmind.net/xmleditor/_download/list- \$\{XXE_VERSION\}\$.xxe_addon](http://www.xmlmind.net/xmleditor/_download/list-$\{XXE_VERSION\}$.xxe_addon) (for example, variable $\{XXE_VERSION\}$ is replaced by "3_5_2" if the version of the running XML Editor is v3.5.2)
- http://www.xmlmind.net/xmleditor/_download/list.xxe_addon

Preference keys: `addonSource`; type: `servers|directory|both`; value: `servers`, or `both` if "Search add-ons in this directory" is also checked. The list of ``servers" is specified by key; `addonServers`; type: list of URLs separated by newline characters (`\n`); default: see above URLs.

Search add-ons in this directory

Zip files containing add-ons may have been downloaded using a Web browser and then copied to a local directory (for example, because you have problems downloading add-ons directly from XXE). In such case, check this box and specify in the text field below it this local directory.

Note that this directory may also contain unzipped add-ons (not recommended though). This is useful because sometimes Web browsers have the bad habit of automatically unzipping the downloaded Zip files.

Default: not checked. No local directory containing Zip files.

Preference keys: `addonSource`; type: `servers|directory|both`; value: `directory`, or `both` if "Download add-ons from these servers" is also checked. The local directory is specified by key: `addonDir`; type: the file-name of an existing directory; default `none`.

Install add-ons in the user's preferences directory

Install all types of add-ons in `XXE_user_preferences_dir/addon/`, where `XXE_user_preferences_dir` is:

- `$HOME/.xxe4/` on Linux.
- `$HOME/Library/Application Support/XMLmind/XMLEditor4/` on the Mac.
- `%APPDATA%\XMLmind\xMLEditor4\` on Windows 2000, XP, Vista.

Example: `C:\Documents and Settings\john\Application Data/XMLmind\xMLEditor4\` on Windows 2000 and XP. `C:\Users\john\AppData\Roaming/XMLmind\xMLEditor4\` on Windows Vista.

This radiobutton is disabled (grayed) if you have insufficient privileges to create files in this directory.

Default: not checked.

Preference key: `addonInstallDir`; type: `user|system|best`; value: `user`.

Install add-ons in XXE installation directory

Install all types of add-ons in `XXE_install_dir/addon/`, where `XXE_install_dir` is, for example, on Windows, `C:\Program Files/XMLmind_XML_Editor`.

This radiobutton is disabled (grayed) if you have insufficient privileges to create files in this directory.

Default: not checked.

Preference key: `addonInstallDir`; type: `user|system|best`; value: `system`.

Installation directory depends on the add-on

Add-ons which depend on a specific version of XXE are installed in `XXE_install_dir/addon/`, other add-ons are installed in `XXE_user_preferences_dir/addon/`.

This way, the add-ons which depend on a specific version of XXE, are *automatically uninstalled* when, in the future, you'll upgrade the application. The other add-ons, which do not on a specific version of XXE, are *not uninstalled* when, in the future, you'll upgrade the application.

In practice, the following add-ons are installed in `XXE_install_dir/addon/`:

- XSL-FO processor plug-ins.
- Image toolkit plug-ins.
- Non-XML format plug-ins.
- Virtual drive plug-ins.
- Configurations, customizing XXE for a given document type, which include custom commands written in Java™.

On the other hand, the following add-ons are installed in `XXE_user_preferences_dir/addon/`:

- Translations of XXE to languages other than English.
- Spell checker dictionaries other than the English one.
- Configurations, customizing XXE for a given document type, which don't include custom commands written in Java™.

Note that this radiobutton is disabled (grayed) if you have insufficient privileges to create files in the `XXE_user_preferences_dir/addon/` directory or in the `XXE_install_dir/addon/` directory, that is, when there is no possible choice for the installation directory.

Default: checked.

Preference key: `addonInstallDir`; type: `user|system|best`; value: `best`.

8.5.10. 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.

This font size is also used to compute the base font size of the tree view. If you need to change the base font size of styled views, you need to use a different option: see Default font size [65] in View Options [65].

Default: default base font size of Java™ applications.

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

Text anti-aliasing

If this toggle is checked, text looks much nicer at the expense of rendering speed.

It is strongly recommended to turn this option off if you run XXE on an old computer. Text anti-aliasing is quite CPU-intensive and therefore makes XXE sluggish on old machines.

When used with Java™ 1.5+, the whole GUI of XXE is anti-aliased. When used with Java™ 1.4, only the text of the document views is anti-aliased.

Note that turning on this option is not needed on the Mac where anti-aliasing is part of the Aqua Look&Feel.

Preference key: `textAntiAliasing`; type: `boolean`; default: `true`. No effect on the Mac.

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/Unix, where the cross-platform look and feel (Metal) is used.

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

Automatically reopen last opened document

This option is examined just after XXE is started, when no documents to be opened have been explicitly specified (e.g. by double-clicking on an XML file or by using command line options). When turned on, this option forces XXE to reopen the last document opened during the preceding editing session.

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

See also the `-last [91]` command line option for a handy alternative.

8.5.10.1. Features Options

Features which are not useful in the general case are hidden by default. You need to check the toggle corresponding to a feature and then restart XXE in order to enable this feature.

Enable the Include Tool

Check this toggle to enable `Edit → Reference → Replace By Reference`, `Edit → Reference → Insert Reference Before`, `Edit → Reference → Insert Reference`, `Edit → Reference → Insert Reference After` and the Include tool.

Feature name: `IncludeTool`.

Enable the Integrated Spreadsheet Engine

Check this toggle to enable `Tools → Spreadsheet` and the `Options → Preferences, Tools|Spreadsheet` group of options.

Feature name: `Spreadsheet`.



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

Enable the Developer Tools

Check this toggle to enable `Options → Reload All Configurations` and `Help → Mouse and Key Bindings`.

Feature name: `DeveloperTools`.

Preference key: `features`; type: list of feature names (`IncludeTool`, `Spreadsheet`, etc) separated by newline characters; default: empty list.

8.5.11. Advanced options

8.5.11.1. 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 `XXE_user_preferences_dir/ser/`. XXE user preferences directory is:

- `$HOME/.xxe4/` on Linux, Mac, and more generally, on Unix.
- `%APPDATA%\XMLmind\XMLEditor4\` on Windows 2000, XP, Vista.

Example: `C:\Documents and Settings\john\Application Data\XMLmind\XMLEditor4\` on Windows 2000 and XP. `C:\Users\john\AppData\Roaming\XMLmind\XMLEditor4\` on Windows Vista.

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.

8.5.11.2. Proxies options

Note

This preference sheet is absent when XMLmind XML Editor has been started using Java™ Web Start. In such case, proxy servers must be configured using the Java™ Control Panel.

A proxy server is a service which allows clients such Web browsers or XMLmind XML Editor to make indirect network connections to other HTTP, FTP, etc, servers. A common proxy server is a caching Web proxy. This proxy provides a nearby cache of Web pages and files available on remote Web servers, allowing clients to access them quickly and reliably.

That's why, if you have problems listing or downloading available add-ons using Options → Install Add-ons, may be your organization uses a proxy server to make connections to the Internet and in such case, this proxy server needs to be declared using this preference sheet.

Direct connection to the Internet

Do not use the services of proxy servers, if any. Make direct connections to the Internet.

Default: checked if Java™ 1.4 is used to run XXE; otherwise not checked.

Preference key: `proxySettings`; type: `none|system|application`; value: `none`.

Use system settings

Use what has been specified in the "control panel" of your desktop/operating system. For example, on Windows, you'll find these settings in Control Panel > Internet Options > Connections tab > LAN Settings.

Only available if Java™ 1.5+ is used to run XXE. Only works on Windows and on Linux/Gnome.

Default: checked if Java™ 1.5+ is used to run XXE; otherwise not checked and *disabled*.

Preference key: `proxySettings`; type: `none|system|application`; value: `system`.

Manual proxy configuration

Specify which proxy server to use for which network protocol. Please ask your network administrator to help you fill this form.

Default: not checked.

Preference key: `proxySettings`; type: `none|system|application`; value: `application`.

HTTP proxy

Specifies the host name of the proxy server to use when HTTP connections are made.

Default: none.

Preference key: `httpProxyHost`; type: host name or host address; default: none.

(HTTP proxy) Port

Specifies the port number of the proxy server to use when HTTP connections are made.

Default: 80.

Preference key: `httpProxyPort`; type: `int`; default: 80.

HTTPS proxy

Specifies the host name of the proxy server to use when HTTPS connections are made.

Default: none.

Preference key: `httpsProxyHost`; type: host name or host address; default: none.

(HTTPS proxy) Port

Specifies the port number of the proxy server to use when HTTPS connections are made.

Default: 443.

Preference key: `httpsProxyPort`; type: `int`; default: 443.

FTP proxy

Specifies the host name of the proxy server to use when FTP connections are made.

Default: none.

Preference key: `ftpProxyHost`; type: host name or host address; default: none.

(FTP proxy) Port

Specifies the port number of the proxy server to use when FTP connections are made.

Default: 80.

Preference key: `ftpProxyPort`; type: `int`; default: 80.

SOCKS proxy

Specifies the host name of the SOCKS V4 or V5 proxy server to use when network connections of any sort are made.

Default: none.

Preference key: `socksProxyHost`; type: host name or host address; default: none.

(SOCKS proxy) Port

Specifies the port number of the SOCKS V4 or V5 proxy server to use when network connections of any sort are made.

Default: 1080.

Preference key: `socksProxyPort`; type: `int`; default: 1080.

Authenticate SOCKS user

Check this toggle if the SOCKS proxy server of your organization requires you to authenticate yourself before using its services.

Default: not checked.

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

(SOCKS) Username

Your name as a user of the SOCKS proxy server.

Default: the login name of the user.

Preference key: `socksUser`; type: `string`; default: the login name of the user.

(SOCKS) Password

Your password as a user of the SOCKS proxy server.

Default: the empty string.

Preference key: `socksPassword`; type: *encoded* string; default: the empty string.

No proxy for

When XMLmind XML Editor runs on these hosts, do not the services of proxy servers, if any. Make direct connections to the Internet.

Default: `"localhost 127.0.0.1"`.

Preference key: `nonProxyHosts`; type: list of host names, host addresses and domain names (e.g. `.acme.com`) separated by spaces; default: `"localhost 127.0.0.1"`.

8.6. The "Install Add-ons" dialog box

This dialog box allows to download and install and/or upgrade and/or uninstall one or more add-ons. This dialog box is not available if XXE has been deployed using Java™ Web Start.

In order to uninstall an add-on, select the Uninstall tab and click on the checkbox of the add-on (looks like this , when clicked looks like this). It is also possible to select the rows of the add-on table by using the Up and Down arrows and to toggle the state of the corresponding checkbox by pressing the space bar.

In order to install or upgrade an add-on, select the Install tab and click on the checkbox of the add-on.

The checkbox of an *upgrade* looks like this . In such case, there is no need to also explicitly uninstall the corresponding add-on because this is done automatically for you. (However it is harmless to do so.)

Important

XXE integrated add-on manager assumes that proxy servers, if any, have been properly specified using the Proxies section [75] of the Preferences dialog box.

Tip

The rows of the Uninstall add-on table have a light yellow background when the corresponding add-on is installed in the user preferences directory, and a light blue background when the corresponding add-on is installed in XXE installation directory.

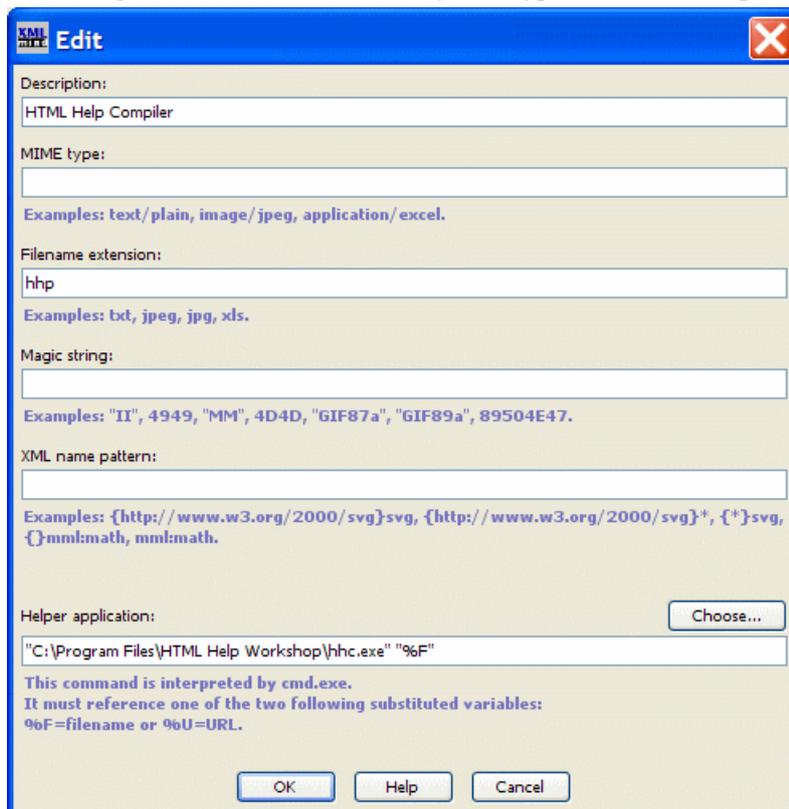
In both cases, a row is disabled (grayed) if you have insufficient privileges to uninstall the corresponding add-on.

Tip

If a problem occurs when listing the add-ons available for download and/or when uninstalling/installing the add-ons, use the "Show Message Log" button of the status bar, category "Install Add-ons", to see logged error messages.

8.7. The "Helper Application Editor" dialog box

This dialog box allows to view or modify a file type listed in the Helper applications preferences sheet [69].



A file type is specified by *at least one* of the following characteristic:

MIME type

The official (or just well-known) formal name of the file type. Generally returned by Web servers to their client Web browsers. Non-registered MIME types typically start with string "application/x-".

A MIME type may end with a wildcard. Example: "image/*" matches "image/gif", "image/jpeg", etc.

Examples: text/plain, image/jpeg, application/excel, application/x-java-help-index.

Filename extension

If the filename or URL of a file ends with specified ".*extension*", then this file is detected as having this file type.

An extension may or may not start with a dot. This is unimportant because, in all cases, a leading dot would be automatically stripped.

Examples: txt, jpeg, jpg, xls.

Magic string

For some file formats, the first bytes of a file are always the same and therefore, can be considered as being the *signature* of this file type.

If a file starts with specified first bytes, then this file is detected as having this file type. This type of detection is supposed to work like magic, hence the name: "magic string".

A magic string may be specified by a the hexadecimal representation of a sequence of bytes (example, one of the two TIFF magic strings: 4949) or by a *quoted* sequence of ASCII characters (same example, one of the two TIFF magic strings: "II").

Examples: TIFF: "II" or 4949, "MM" or 4D4D; GIF: "GIF87a", "GIF89a"; PNG: 89504E47; PDF: "%PDF-".

XML name pattern

If the root element of an XML file has a name which matches specified pattern, then this XML file is detected as having this file type.

An XML name pattern follows this syntax:

```
( '{' namespace_URI? '}' )? local_part
```

One of *local_part* or *namespace_URI* may be equal to wildcard "*"

Examples: {*}svg, {http://www.w3.org/1998/Math/MathML}:math.

Each file type has an associated helper application. This helper application is assumed to be able to open files detected as having this type. A helper application may be a viewer or an editor.

Description

Description of the file type. Not mandatory, just recommended. This text is displayed in the File types list of the Helper applications preferences sheet [69].

MIME type

One or more MIME types (see definition [78] above) separated by spaces.

Filename extension

One or more extensions (see definition [79] above) separated by spaces.

Magic string

One or more magic strings (see definition [79] above) separated by spaces.

XML name pattern

One or more XML name patterns (see definition [79] above) separated by spaces.

Helper application

This field must contain a command line interpreted by the native shell of the platform: `cmd.exe` on Windows and `/bin/sh` on the Mac and on Unix.

This command line must reference one of these two substituted variables: %U and %F. In principle, %U is replaced by the URL of the file to be opened by the helper application and %F is replaced by a filename. In practice, %U is just a *hint* meaning: the helper application can open URLs as well as filenames.

The Choose button displays the standard file chooser in order to specify an application (e.g. a .exe or a .bat file on Windows). String " %F" is automatically appended to the chosen application.

Helper applications on the Mac

When an *application* (that is, a folder having a name ending with hidden suffix ".app", containing a package bundle) has been selected by the user, the Choose button automatically prepends:

- for a version older than Mac OS X 10.5 (e.g. Tiger): "open -a",
- starting from Mac OS X 10.5 (Leopard): "open -W -n -a".

Example: "open -W -n -a /Applications/Inkscape "%F".

Options "-w -n" mean: start a new instance of the application and wait until this instance has exited. These options are required when the helper application is used to edit the content of an element, the content of an attribute or the whole document.

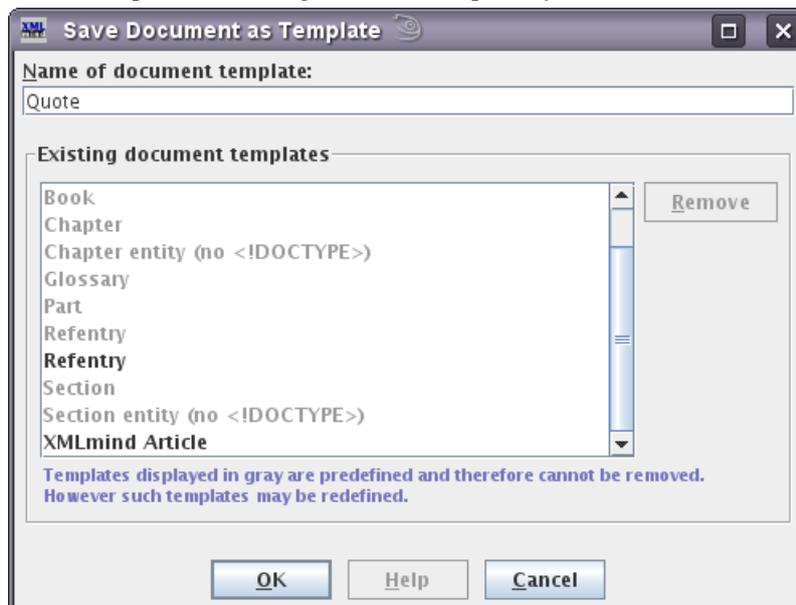
In practice, this means that, on Mac OS X versions older than 10.5, a helper application can only be used for viewing purposes and this, even if the helper is an editor.

8.8. The "Save Document As Template" dialog box

This dialog box is displayed by menu item Options → Customize Configuration → Save Document as Template [29]. Basically, it allows the user to specify a name for the newly created document template.

The name of a document template may contain any character, including spaces.

The name of a document template may be identical to the name of one of the existing document templates. This allows to replace an existing document template by a new one.



In the above screen shot, we can see that:

- The user is creating a new document template called "Quote".

- Previously, the user has replaced predefined template called "Refentry" by a "Refentry" document template of its own.
- Previously, the user has created another document template called "XMLmind article".

8.8.1. Removing a previously defined document template

This dialog box also allows to remove user-defined document templates. In order to do this, simply click on a template in the "Existing document templates" list and then click on the Remove button.

Note that this operation takes immediate effect: clicking on the Cancel button of the "Save Document As Template" dialog box will not cancel the deletion of the template¹¹.

You can only remove the document templates you have created. You cannot remove predefined (grayed) document templates.

In the above screen shot, notice how removing user-defined template called "Refentry" would restore the predefined template having the same name.

8.9. The "Save Selected Element as Template" dialog box

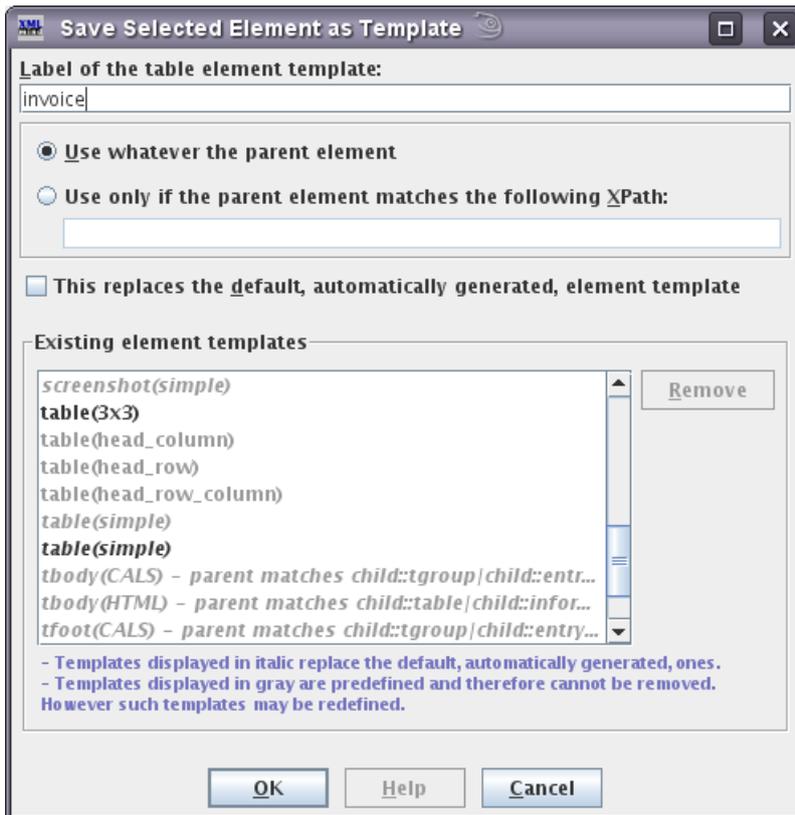
This dialog box is displayed by menu item Options+Customize Configuration → Save Selected Element as Template [29]. Basically, it allows the user to specify a *label* for the newly created element template.

Giving a label to an element template is needed to support different templates for the same element type. For example, in a DocBook document, select a `table` element and use Options+Customize Configuration → Save Selected Element as Template to label it as "L1". Then select another `table` element and use the same menu item to label it as "L2". This gives you two `table` element templates, one called "L1" and the other called "L2".

The label of an element template must be a valid `NMTOKEN`. It cannot contain whitespace.

The element name and label of a template may be identical to the element name and label of one of the existing templates. This allows to replace an existing template by a new one.

¹¹The Cancel button of the "Save Document As Template" dialog box just cancels the action of creating a new template.



In the above screen shot, we can see that:

- The user is creating a new `table` element template called "invoice".
- After creating this template, it will be displayed in the "Existing element templates" list as "table(invoice)", "table" being the element name of the template and "invoice" being its label.
- Previously, the user has created another `table` element template labelled "3x3".
 - Previously, the user has replaced predefined "table(simple)" template by a "table(simple)" template of its own.

8.9.1. Advanced options

Use only if the parent element matches the following XPath

Normally an element template is suggested by the Edit tool whatever the parent element where this template is to be inserted. But in some cases, generally with W3C XML schemas and with RELAX NG schemas and not with DTDs, you'll want the template to be usable only in certain contexts.

Example: your schema defines two `title` elements: a *job title* which is a possible child element of `person` and `author` elements and the optional title (i.e. a *caption*) of `table` and `figure` elements.

You intend to define several job title templates. Example: element template "title(arch)" is defined as `<title>Software Architect</title>`.

Of course, you don't want to see "title(arch)" listed by the Edit tool when you are about to give a title to a `table`.

In such case:

1. Click on the "Use only if the parent element matches the following XPath" radio button.
2. Type XPath expression "`person|author`" in the text field below the radio button.

This means: suggest "title(arch)" but only when the template is to be inserted in a `person` or an `author` element.

Note that a very small subset of XPath is supported here:

XPath 1.0 subset

The XPath 1.0 subset is the one defined in "XML Schema Part 1: Structures, Identity-constraint Definitions", except that absolute XPaths (`/foo/bar`, `//bar`, etc) are also supported.

```
XPath      ::= Path ( '|' Path ) *
Path       ::= ( '/' | '//') ? ( Step ( '/' | '//') ) * ( Step | '@' NameTest )
Step       ::= '.' | NameTest
NameTest   ::= QName | '*' | NCName ':' '*'
```

Both abbreviated syntax and non-abbreviated syntax are supported.

This replaces the default, automatically generated, element template

This option is best explained by an example.

Out of the box, if you insert a `table` element in a DocBook document, in fact, you insert predefined `table` template called "table(simple)".

The "table(simple)" template has two rows and two columns which makes it much more useful than the default, automatically generated, `table` template which just contains a single cell¹².

Note that when a *single* named template such as "table(simple)" replaces the default, automatically generated, element template, the Edit tool does not display its label. For example, in the case of the DocBook `table`, the Edit tool displays "table"¹³ and not "table(simple)".

Now, what if you prefer to use a `table` having 3 rows and 3 columns plus a `thead` header as your default `table`?

1. Create the table by inserting the rows, cells and header you want.
2. Select this table.
3. Use Options → Customize Configuration → Save Selected Element as Template.
4. The "Existing element templates" list shows you that a predefined template already replaces the default, automatically generated, one. You can see this because "table(simple)" is displayed in italic.

You need to label your own template like the existing one in order to replace it. Therefore specify "simple" in the Label of the element template field.

5. Click on check box "This replaces the default, automatically generated, element template".

8.9.2. Removing a previously defined element template

This dialog box also allows to remove user-defined element templates. In order to do this, simply click on a template in the "Existing element templates" list and then click on the Remove button.

Note that this operation takes immediate effect: clicking on the Cancel button of the "Save Selected Element As Template" dialog box will not cancel the deletion of the template¹⁴.

¹²The default, automatically generated, element template is the simplest, valid, element allowed by the schema.

Most of the time this heuristic gives good results. However in some cases, the default, automatically generated, element template, is, well, too simple to be useful.

¹³There is no ambiguity and this is shorter to type.

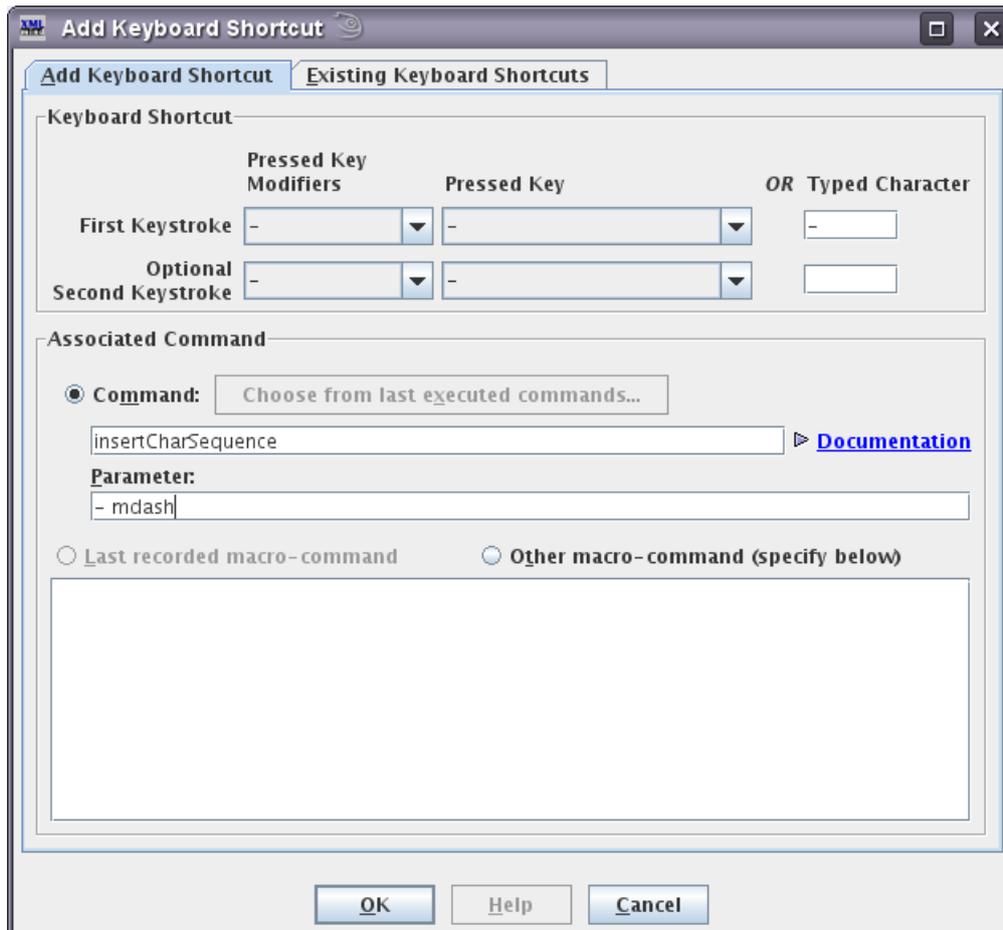
¹⁴The Cancel button of the "Save Selected Element As Template" dialog box just cancels the action of creating a new template.

You can only remove the element templates you have created. You cannot remove predefined (grayed) element templates.

In the above screen shot, notice how removing user-defined "table(simple)" template would restore the predefined template having the same element name and label.

8.10. The "Add Keyboard Shortcut" dialog box

This dialog box is displayed by menu item Options → Customize Configuration → Add Keyboard Shortcut [30]. It allows the user to specify a keyboard shortcut that will invoke a predefined command, last recorded macro-command or a user-defined macro-command.



In the above screen shot, the user adds a shortcut for command "insertCharSequence" with parameter "- mdash". This command will be invoked after character "-" (dash) is typed.

Procedure 2. Adding a keyboard shortcut

1. Specify a keystroke. A keystroke is either:

- A *typed character* specified the Typed Character field.

A typed character is any character you can type using any combination/sequence of pressed keys. A typed character often does not correspond to a key of your keyboard. For example, on an US keyboard, there is no \$ key. You need to press on key **4** and on key **Shift** at the same time in order to type the '\$' character.

- OR a *pressed key* specified using the Pressed Key combo box.

A pressed key must correspond to a key actually found on your keyboard. For example, do not choose key \$ on an US keyboard because there is no such key. On the other hand, you may choose it on a French keyboard which has a \$ key¹⁵.

You'll often want to also specify the modifier keys (**Ctrl**, **Shift**, etc) which must be pressed at the same time as the main key specified in the Pressed Key combo box. Any combination of modifier keys may be chosen using the Pressed Key Modifiers combo box.

2. Optionally specify a second keystroke. In such case, the command will be invoked by the sequence: first keystroke immediately followed by second keystroke.
3. Specify the command invoked by your custom shortcut. There are three possible options for this:

Command

The command invoked by your custom keyboard shortcut is a predefined command (documented in Chapter 6, *Commands written in the Java™ programming language in XMLmind XML Editor - Commands*).

You may choose one of the last executed commands by using the list displayed by the "Choose from last executed command" button.

Last recorded macro

The command invoked by your custom keyboard shortcut is the last recorded macro-command (see Tools → Record Macro [26]).

Other macro-command

The command invoked by your custom keyboard shortcut is a custom macro-command that you'll type in the text area below the "Other macro-command" radio button.

It is recommended to restrict yourself to specifying one of the last executed commands or last recorded macro. Attempting to use the other options is harmless, but unless you know what you are doing (for example, because you have posted a support request to the <xmlmind-support@xmlmind.com> public mailing list), the commands specified the hard way will probably not work.

However, if you decide to follow the hard way, you must know that:

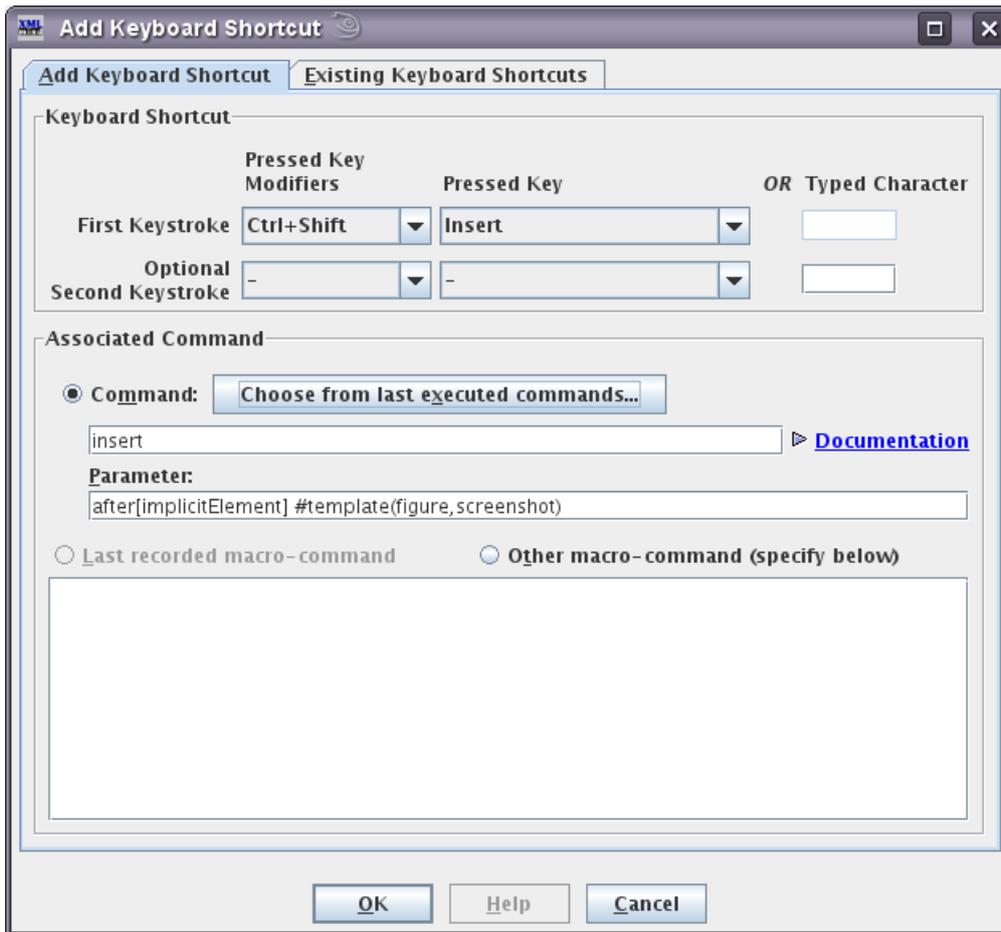
- The Command field, allowing to specify the name of a command, supports auto-completion.
- Clicking on the Documentation link will open in the Web browser the reference manual in *XMLmind XML Editor - Commands* describing the command of interest¹⁶. This is almost always needed in order to specify in the Parameter field a valid parameter for the command.
- If you type a custom macro-command in the text area, do not bother declaring namespace prefixes. Consider that "http://www.xmlmind.com/xmlmind/schema/configuration" is declared as being the default namespace, that "cfg" is declared as a prefix for namespace "http://www.xmlmind.com/xmlmind/schema/configuration" and that all namespace prefixes declared in the document being edited (see Tools → Declare Namespace [23]) are in scope also here.

4. Click OK.

Other example:

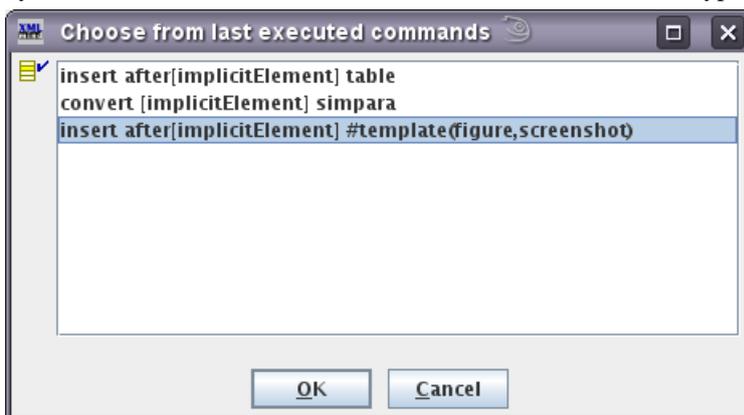
¹⁵Suprising, isn't it?

¹⁶This requires an access to the Internet if XMLmind XML Editor is deployed using Java™ Web Start.



In the above screen shot, the user adds a shortcut for command "insert after[implicitElement] #template(figure,screenshot)". This command will be invoked when the Insert key will be pressed at the same time as the Control and Shift keys.

Previously, by using the Edit tool, the user has inserted in the document being edited a `figure` element containing a `screenshot` child element¹⁷. Having done this now allows her/him to pick this command from the list displayed by the "Choose from last executed command" button, instead of typing the command name and its parameter.



Note

Many keystrokes may not be used to define custom keyboard shortcuts. Clicking on the "Existing Keyboard Shortcuts" tab will give you an idea of this fact.

¹⁷She/he picked `figure(screenshot)` from the list of elements suggested by the Edit tool.

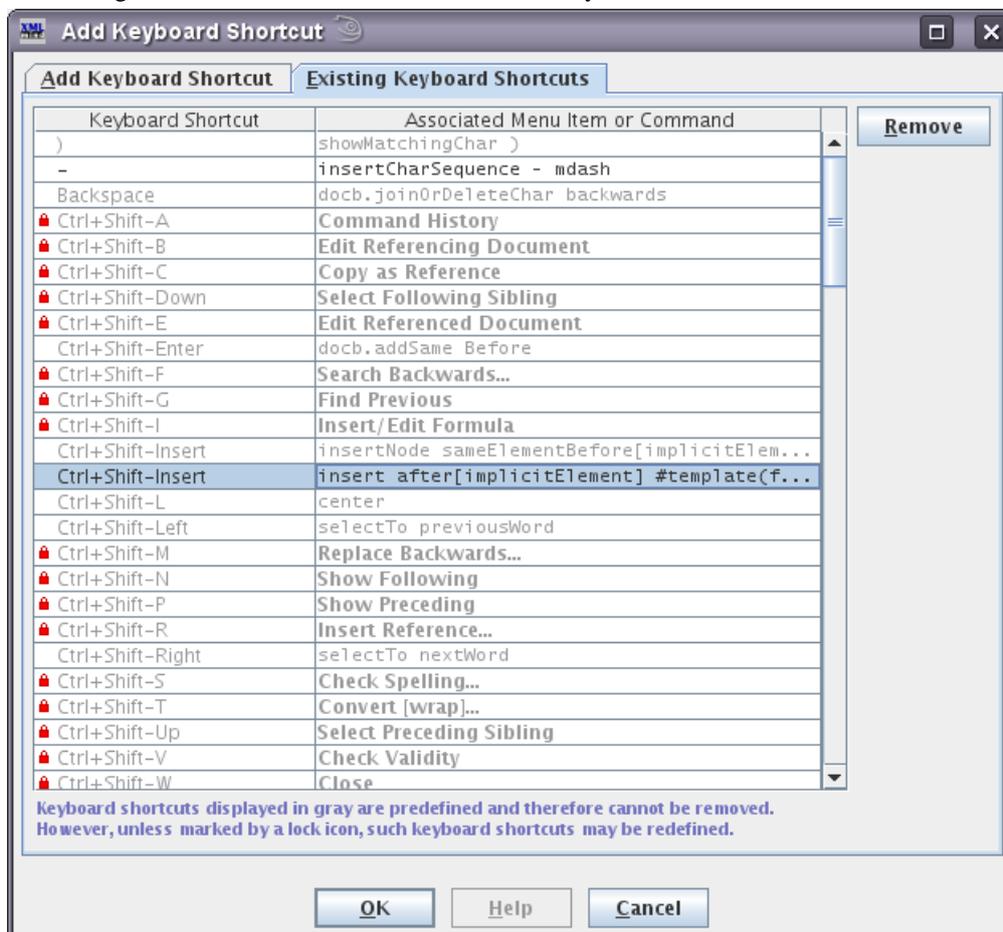
Examples:

- You may not use **Ctrl+O** because this keystroke is reserved for menu item File → Open.
- You may not use **Ctrl+I** because this keystroke is reserved for menu item Edit → Insert.
- You cannot use single keystroke **Esc** because this would prevent existing keyboard shortcuts **Esc g** (bound to menu item Select → Find Element), **Esc x** (bound to menu item Tools → Execute) , etc, from working¹⁸.
- You cannot use keystroke **Home**, for example followed by a second **Home** keystroke, because existing keyboard shortcut **Home** (bound to command "moveDotTo lineBegin") would prevent your custom shortcut from working.

Moreover certain keystrokes are reserved by Java™ (e.g. Tab, **Alt**) or by the operating system (e.g. there are loads of these on the Mac). Attempting to use these keystrokes for your custom keyboard shortcuts will almost certainly not work.

8.10.1. Removing a previously defined keyboard shortcut

This dialog box also allows to remove user-defined keyboard shortcuts.



In the above screen shot, notice how removing user-defined keyboard shortcut **Ctrl+Shift+Insert** (invoking command "insert after[implicitElement] #template(figure,screenshot)") would restore the predefined keyboard shortcut (invoking command "insertNode sameElementBefore[implicitElement]").

¹⁸Using **Esc** followed by a second, carefully chosen, keystroke should work fine.

Procedure 3. Removing a keyboard shortcut

1. Click on the "Existing Keyboard Shortcuts" tab.
2. Select the keyboard shortcut to be removed by clicking on it.

You can only remove the keyboard shortcuts you have created. You cannot remove predefined (grayed) keyboard shortcuts.

3. Click on the Remove button.

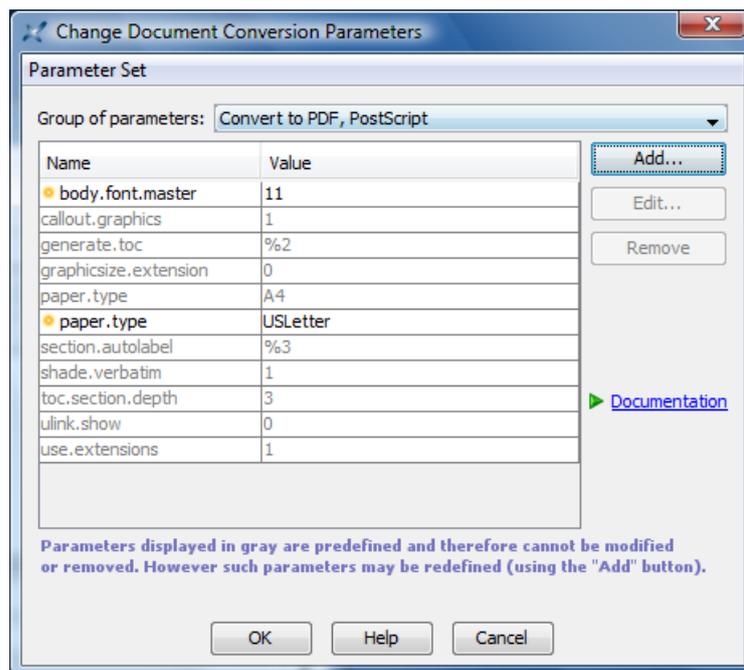
Note that this operation takes immediate effect: clicking on the Cancel button of the "Add Keyboard Shortcut" dialog box will not cancel the deletion of the keyboard shortcut¹⁹.

8.11. The "Change Document Conversion Parameters" dialog box

This dialog box is displayed by menu item Options → Customize Configuration → Change Document Conversion Parameters [30]. This dialog box allows to specify the XSL style sheet parameters (e.g. `paper.type=USLetter`) used when converting the document being edited to other formats (e.g. PDF, HTML, etc).

Procedure 4. Adding a user-defined parameter

1. Select the conversion for which you want to specify parameters by using the "Group of parameters" combo box.



In the above screen shot, the user has selected the conversion of DocBook documents to PDF or PostScript.

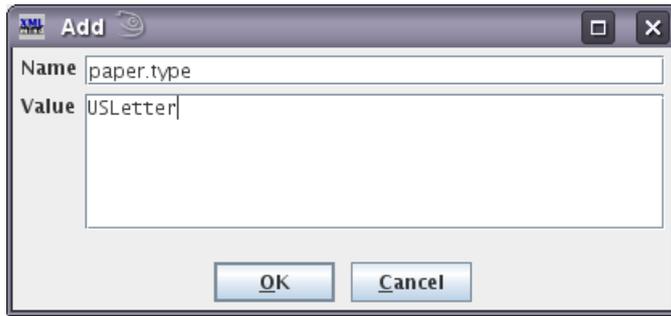
2. Click on the Add button.

Tip

If, for example, you want to redefine `paper.type` from `A4` to `USLetter` as shown in the above screen shot. Select the `paper.type` parameter by clicking on it and then click on the Add button. This will spare you the effort of retyping `paper.type` as the name of the parameter you want to add.

¹⁹The Cancel button of the "Add Keyboard Shortcut" dialog box just cancels the action of creating a new keyboard shortcut.

3. Specify the name of the parameter and its value in the dialog box displayed by the Add button and then click OK.



The value of a parameter may contain whitespace, including newline characters.

4. Click OK if you are done or use "Group of parameters" combo box and the Add, Edit and Remove buttons to specify more parameters.

In the above screen shot:

- The user has added parameter `body.font.master=11`. (The default value of this parameter, as specified in the XSL style sheet, is 10.)

Notice the star which precedes newly added and newly modified parameters.

- The user has redefined parameter `paper.type` to `USLetter`.

Notice how the user-defined parameter follows the predefined, grayed, parameter.

Tip

The Documentation link is supposed to open in your favorite Web browser the reference manual of the XSL style sheet parameters.

Therefore, If you select a parameter by clicking on it and then click on the Documentation link, you should be able to read the documentation of this parameter.

However, this is not guaranteed to work because:

1. This generally requires an access to the Internet.
2. This documentation is not always available. For example, there is no documentation about the XHTML XSL style sheets yet. In such case, you need to send a support request to the xmlmind-support+xmlmind.com public mailing list to learn how the document conversion process can be parametrized.

Procedure 5. Modifying a user-defined parameter

1. Select the conversion for which you want to specify parameters by using the "Group of parameters" combo box.
2. Select the parameter to be modified by clicking on it.
3. Click on the Edit button.

Predefined, grayed, parameters cannot be modified.

4. Specify the value of the parameter in the dialog box displayed by the Edit button and then click OK.

Tip

It also possible to change the *name* of the parameter here. Doing this will simply add a new parameter or replace an existing one.

In fact, the Add and Edit buttons may almost be used interchangeably.

5. Click OK if you are done or use "Group of parameters" combo box and the Add, Edit and Remove buttons to specify more parameters.

Procedure 6. Removing a user-defined parameter

1. Select the conversion for which you want to specify parameters by using the "Group of parameters" combo box.
2. Select the parameter to be removed by clicking on it.
3. Click on the Remove button.

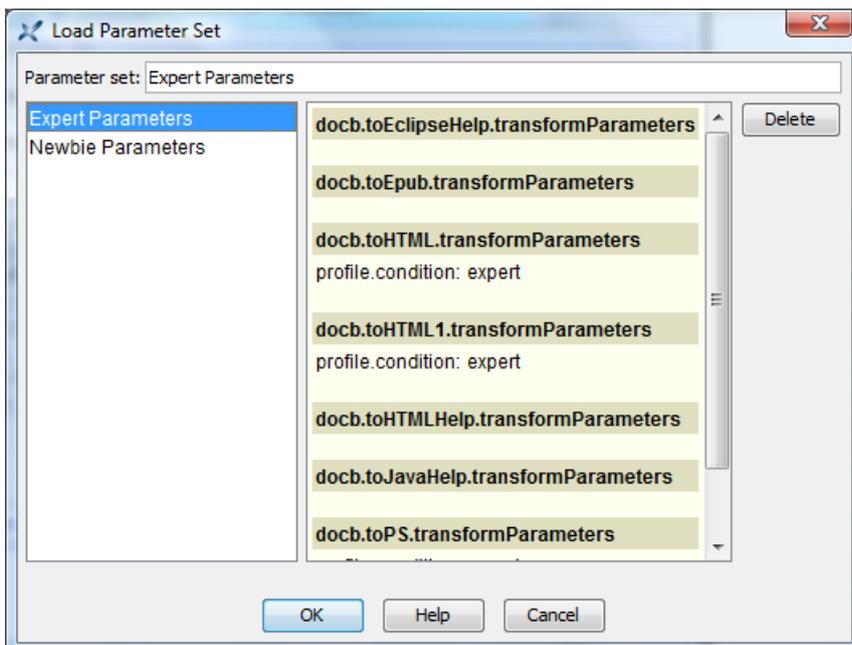
Predefined, grayed, parameters cannot be removed.

4. Click OK if you are done or use "Group of parameters" combo box and the Add, Edit and Remove buttons to specify more parameters.

8.11.1. The "Parameter Set Chooser" dialog box

The "Change Document Conversion Parameters" dialog box has a Parameter Set menu which allows to load and save *named parameter sets*. A parameter set is the set of all the XSL style sheets parameters specified by a user using the "Change Document Conversion Parameters" dialog box. This facility is almost indispensable to deal with the DocBook profiling XSL style sheets. Using such style sheets requires the user to quickly switch from one parameter set (e.g. containing `profile.condition=newbie`) to another (e.g. containing `profile.condition=expert`).

Figure 16. The "Parameter Set Chooser" dialog box



Reset

Removes *all* the parameters currently specified by the user in the "Change Document Conversion Parameters" dialog box. The effect of Reset is thus to restore the stock configuration by removing all the user's customizations.

Example: parameter `paper.type=USLetter` is specified in both the `docb.toPS.transformParameters` and `docb.toRTF.transformParameters` groups. Selecting Reset from the Parameter Set menu means removing these two occurrences of the `paper.type` parameter.

Load

Displays the "Parameter Set Chooser" dialog box to let the user specify an existing parameter set. The contents of this parameter set is then used to *overwrite* all the parameters currently specified by the user in the "Change Document Conversion Parameters" dialog box.

Example: the user loads a parameter set called "USLetter" containing just `paper.type=USLetter` in both the `docb.toPS.transformParameters` and `docb.toRTF.transformParameters` groups. Selecting Load from the Parameter Set menu means removing all the currently specified parameters from all the parameter groups and then adding `paper.type=USLetter` to both the `docb.toPS.transformParameters` and `docb.toRTF.transformParameters` groups.

Save

Displays the "Parameter Set Chooser" dialog box to let the user specify the name of a parameter set. The integral contents of the "Change Document Conversion Parameters" dialog box at the save time is then copied to this parameter set.

A. Command line options

1. Command line

```
xxe [advanced_option]*
( [-new config template save_file_or_URL | file_or_URL_to_be_opened]* | -last )
```

Options:

`-new config template save_file`

Command-line equivalent to using File → New.

`config`

Specifies the (case-insensitive) name of a configuration. Examples: DocBook, docbook, "DocBook v5+", "XHTML Strict", "xhtml transitional".

`template`

Specifies the (possibly localized) name of a document template. Example: "Seite" ("Page" in German).

Alternatively, you can specify the basename (without any extension) of the file containing the document template. Doing this should work whatever your locale. Example: "page_strict".

Use "-" to specify the first available document template. This option is mainly useful with configurations having a single document template such as "slides".

`save_file_or_URL`

Specifies the filename or URL of the newly created document. Note that specifying such filename does not create the corresponding save file. You'll have to use File → Save or File → Save As to actually save the newly created document.

Use "-" to let XMLmind XML Editor choose this filename for you (as it does it when you use File → New).

Examples:

- Create a new Slides¹ presentation. The filename of the new presentation is `MyPresentation.xml`. This file will be found in the current working directory.

```
-new slides - MyPresentation.xml
```

- Create a new "XHTML Strict" page and let XXE choose the filename for you:

```
-new "XHTML Strict" Seite -
```

or:

```
-new "xhtml strict" page_strict -
```

- Create a new DocBook section in file `sections/s4.xml` (relative to the current working directory):

```
-new docbook section sections/s4.xml
```

-last

Forces XXE to reopen the last document opened during the preceding editing session. Note that this option cannot be used when one or more *file_or_URL_to_be_openeds* have been specified.

See also the "Automatically reopen last opened document [74]" option for a handy alternative.

Advanced options:

-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 Preferences dialog box [56].

The description of each option includes a short description of the corresponding preference key. For example, in the case of the Encoding [58] option, the preference key is `encoding`. Example:

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

Note

XXE user preferences directory is:

- `$HOME/.xxe4/` on Linux.
- `$HOME/Library/Application Support/XMLmind/XMLEditor4/` on the Mac.
- `%APPDATA%\XMLmind\xMLEditor4\` on Windows 2000, XP, Vista.

Example: `C:\Documents and Settings\john\Application Data\xMLmind\xMLEditor4\` on Windows 2000 and XP. `C:\Users\john\AppData\Roaming\xMLmind\xMLEditor4\` on Windows Vista.

-putprefs *property_file_or_URL*

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

```
xxe -putprefs /etc/xxe/preferences.properties
```

-delpref *key*

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

¹Requires installing the corresponding add-on.

-auth credentials

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

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

Command-line utility "java -cp xxe.jar com.xmlmind.netutil.SimpleAuthenticatorModule" allows to generate such encoded string. Example: encode string "\n\nDocument Store\n\nvictoria\n\nsecret":

```
/opt/xxe/bin$ java -cp xxe.jar com.xmlmind.netutil.SimpleAuthenticatorModule \
victoria secret - "Document Store"
CgpEb2N1bWVudCBTdG9yZQoKanZpY3RvcmlhCnNlY3JldA==

/opt/xxe/bin$ xxe -auth CgpEb2N1bWVudCBTdG9yZQoKanZpY3RvcmlhCnNlY3JldA== \
http://www.acme.com/docstore/push_up.xml &
```

-open file_or_URL_to_be_opened, -print file_or_URL_to_be_opened

First *file_or_URL_to_be_opened* may be preceded by *-open* or *-print*, which are ignored. This may be useful when XXE is started by Java™ Web Start. Example: "javaws http://www.acme.com/xxe/xxe.jnlp -open /docs/doc.xml".

2. Environment variables

All scripts used to start XXE (that is, *xxe*, *xxe.bat* and *xxe.jstart*) automatically define system properties corresponding to the following environment variables.

Note that, due to limitations in Apple's Java™ launcher, you cannot use the following environment variables on the Mac².

Variable name	Value	Description
XXE_ADDON_PATH	List of directory names separated by semi-colons (;).	All the directories referenced in this list are recursively scanned by XXE during its startup to dynamically discover add-ons. More info in chapter "The lookup phase during XXE startup" of Section 1, "Dynamic discovery of add-ons" in <i>XMLmind XML Editor - Configuration and Deployment</i> .
XXE_GUI	Filename (relative or absolute) or URL of an XXE GUI specification (.xxe_gui) file.	Specifies which GUI to use for newly started XXE. More info in <i>XMLmind XML Editor - Customizing the User Interface</i> .
XXE_USER_PREFERENCES	Filename of a property file	Specifies the location of a user preferences property file different from the default one: <i>XXE_user_preferences_dir/preferences.property</i> . This alternate user preferences property file is created (if needed to)/read from/written to by XXE exactly like the default one.

²This is the case only with *XMLEditor.app*, the application bundle contained in the *.dmg* distribution. If you are a local guru or a consultant you may prefer to download and install the *xxe-*mac.zip* distribution rather than the *xxe-*.dmg* distribution.

After unzipping this archive in a directory of your choice, XMLmind XML Editor may be started using the *XXE_install_dir/bin/xxe* shell script. Unlike *XMLEditor.app* which leverages Apple's Java™ launcher, the *xxe* shell script makes it easy working with environment variables such as *XXE_GUI*, *XXE_ADDON_PATH*, etc.

Variable name	Value	Description
		Note that the hierarchy of directories containing this alternate user preferences property file is also created by XXE when it does not already exist.

3. System properties

In principle, there is no need to use the following, very low-level, system properties.

Property name	Value	Description
XXE_ADDON_DOWNLOAD_LOCATION	List of ".xxe_addon" URLs separated by semi-colons (;).	<p>Clicking on the Reset button of the "Install add-on" preferences of the Preferences dialog box resets the list of download URLs to the value specified by this property.</p> <p>If this property is not set, by default, the list of download URLs is</p> <ul style="list-style-type: none"> • http://www.xmlmind.com/xmleditor/_usercontrib/list.xxe_addon • http://www.xmlmind.net/xmleditor/_download/list-{\$XXE_VERSION}.xxe_addon (for example, variable <code>{\$XXE_VERSION}</code> is replaced by "3_5_2" if the version of the running XML Editor is v3.5.2) • http://www.xmlmind.net/xmleditor/_download/list.xxe_addon
XXE_DAV_TRACE	Any (example: 1; suffice to <i>set</i> this property).	Causes the WebDAV client to print DAV requests and responses on the console.
XXE_DAV_URI_ENCODING_USES_8859_1	Any (example: 1; suffice to <i>set</i> this property).	<p>Used by the WebDAV client.</p> <p>Specifies that the WebDAV server expects and returns URIs where accented characters (to make it simple) are escaped using the customary %<i>HH</i> form but where <i>HH</i> is the hexadecimal code of an ISO-8859-1 character.</p> <p>The standard mandates <i>HH</i> to be UTF-8 codes, not ISO-8859-1 codes.</p> <p>For example, this flag is needed for Apache/mod_dav running on an "old" Linux server where the file system encodes filenames using ISO-8859-1.</p>
XXE_DAV_LOCK_OWNER	A free form string, typically an email address.	<p>Used by the WebDAV client.</p> <p>Specifies which string is used by default to identify you as the owner of a lock.</p>
XXE_NETWORK_TIMEOUT	A number of milliseconds.	<p>Specifies the default timeout used for creating network connections and reading responses. A negative or null number means: wait indefinitely.</p> <p>Used at various places in XMLmind XML Editor, for example by WebDAV and FTP virtual drives.</p> <p>Built-in default value is 120,000ms (2mn).</p>

Property name	Value	Description
XXE_JAVA_IMAGE_TOOLKIT_EXCLUDE	List of file extensions separated by semi-colons (';').	Example: <code>-DXXE_JAVA_IMAGE_TOOLKIT_EXCLUDE="bmp;wbmp"</code> This would prevent the built-in Java™ 1.5 image toolkit from handling BMP images. This is useful in case you find an external image toolkit which handles specified image extensions better than the built-in Java™ image toolkit.
XXE_CATALOG_RESOLVER_VERBOSITY	Strictly positive integer (the larger, the more verbose)	Prints on the console messages tracing the resolution by the XML catalog resolver of system identifiers.
javaws.XXE.*	-	Reserved. Do not use.