XMLmind XML Editor - Creating a TEI Lite Document

Hussein Shafie

XMLmind Software

35 rue Louis Leblanc,
78120 Rambouillet,
France,
Phone: +33 (0)9 52 80 80 37,
Web: www.xmlmind.com/xmleditor
Email: xmleditor-support@xmlmind.com

June 30, 2026

Table of Contents

1. What you'll learn 1
2. XMLmind XML Editor basics 2
    2.1. A strictly validating XML editor 2
    2.2. Basic concepts 3
        2.2.1. The selection modes 6
        2.2.2. The basic editing commands 7
3. Creating a TEI Lite document 9
4. Adding a division 12
    4.1. Specifying the type of a division 12
5. Paragraphs 14
    5.1. Other text blocks 14
6. Lists 16
    6.1. List items 17
7. Figures 20
8. Tables 23
9. Phrase level elements 25
10. Links 27
11. Other features 29

1. What you'll learn

By reading this document, you'll learn to use XMLmind XML Editor (**XXE** for short) to create and modify a TEI Lite document. We'll assume here that you know TEI and XML, but that you don't know anything about **XXE**.

Remember

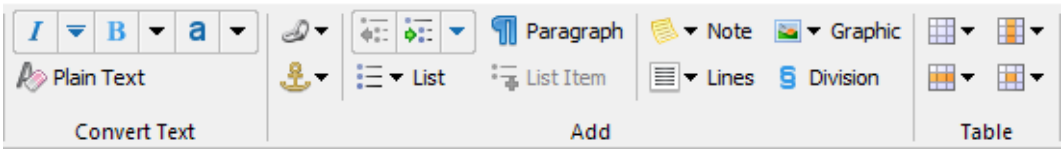
Only the TEI Lite tagset is supported. *Full TEI P5 is **not** supported. Any customization of the TEI tagset other than TEI Lite is **not** supported.*

In terms of document editing, TEI Lite support in XMLmind XML Editor is as extensive as DITA, DocBook or XHTML support. However converting a TEI document to other formats (HTML, PDF, EPUB, DOCX, etc) is currently not supported.

If there is sufficient interest for this work (or if we find a sponsor willing to partially fund this development), we may implement full TEI P5 support and/or TEI document conversion facilities in the future.

Except for first lesson 2. XMLmind XML Editor basics, each lesson will describe only the easiest ways to perform an editing operation (such as adding a list). This generally involves using TEI-specific toolbar buttons rather than the generic **Edit** tool and/or **Attributes** tool.

Figure 1-1. *The TEI-specific toolbar buttons*



Some chapters of this document contain links to online XMLmind XML Editor tutorials (including screencasts). Some other chapters contain links to customization tips explaining how you can easily adapt **XXE** to your needs (e.g. custom element templates). These online tutorials are all based on DocBook 5 examples, however what you'll learn there applies to all document types, including TEI.

Important

TEI Lite support in XMLmind XML Editor is available in the form of an *add-on* called "**TEI Lite configuration**". This add-on is not installed by default. Before starting to read this document and may be, experiment with **XXE**, please use menu entry "**Options|Install Add-on**" to install this add-on.

2. XMLmind XML Editor basics

XMLmind XML Editor (**XXE** for short) is very straightforward to use but does not resemble any other XML editor, even less a word processor. That's why it is important to carefully read and understand this chapter. If you do this, you'll never get stuck while using **XXE**. You may even become very productive at creating TEI documents with it.

2.1. A strictly validating XML editor

XMLmind XML Editor (**XXE** for short) is advertised as being *a strictly validating XML editor*. But what does this mean?

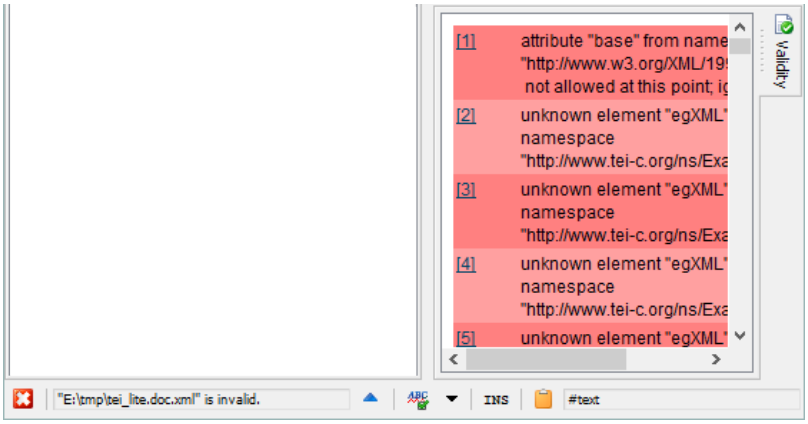
If the document opened in **XXE** is initially valid —in our case, conforms to the TEI Lite grammar— then **XXE** will not let you modify this document in a way which would make it invalid. For example:

- You may not delete the `title` child element found in a `titleStmt` element (because a `titleStmt` element must start with one or more `title` elements).
- You may not insert a division between two paragraphs (this is specified in the content models of `body` and `div`)
- You may not set an `xml:id` attribute to a number (because an ID attribute may not start with a digit).

What about an invalid TEI document created outside XXE?

Of course, you can open an invalid TEI document in **XXE**[1] and still be able to edit this document. When this is the case, you'll see something like this:

Figure 2-1. *An invalid document opened in **XXE***



- The "**Validity state**" indicator shows a blinking red cross .
- The **Validity** tool is brought to front and lists a number of error messages. Clicking the number of an error message selects in the document view the element where the validity error was found.

[1] You can even open a non-well-formed document in **XXE** but in this case, you'll see just the raw XML source of your document.

Remember

Please remember that **XXE** currently only supports TEI Lite and no other TEI tagset. Therefore you may open a perfectly valid TEI document in **XXE** and still get *thousands* of the most severe validity errors. This happens simply because your document uses TEI elements (e.g. the `ab` and `egXML` elements) beyond those found in TEI Lite.

For example, opening the original "`tei_lite.doc.xml`" —C.M. Sperberg-McQueen and Lou Burnard, *TEI Lite: Encoding for Interchange: an introduction to the TEI*— in **XXE** causes the **Validity** tool to list 10 359 error messages!

This blinking red cross means that the document has *the most severe validity errors*[2]. For example, an unknown element was found in the document.

WARNING

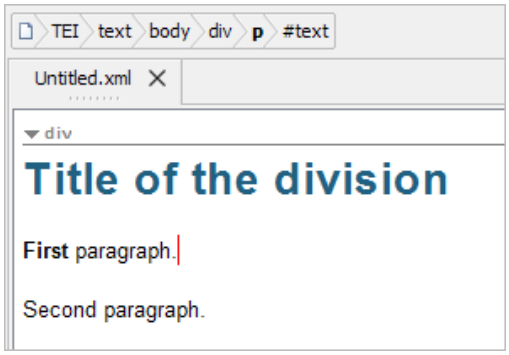
You are expected to fix all these severe validity errors before actually using **XXE** to add content to your document.

If you cannot or do not want to fix all the severe validity errors, then we recommend to stop using **XXE**. Our XML editor *has not been designed to function normally in presence of severe validity errors.*

2.2. Basic concepts

Let's suppose that you want to add a general note (i.e. not a footnote) before first paragraph. How would you do that using XMLmind XML Editor (**XXE** for short)?

Figure 2-2. *Styled view of a sample TEI document*

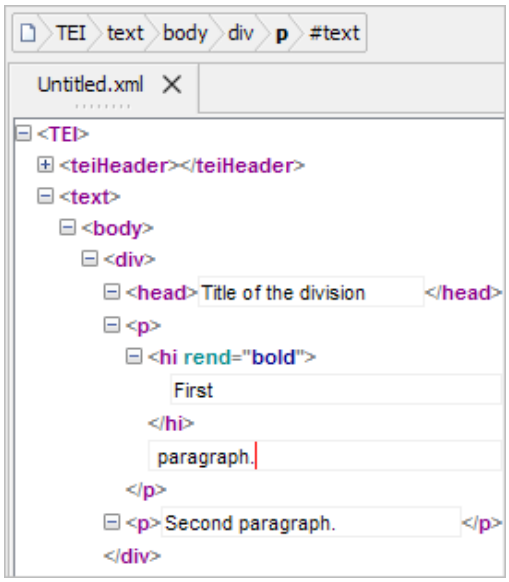


[2] The "**Validity state**" indicator may display several other icons. For example , which means that the document has minor validity errors such as bad cross-references. Working on a document which has bad cross-references is very common, so you should not be alarmed by this situation.

The answer is: first select the first `p` element and then insert a `note` element before the selection. Alternatively, you could select the `head` element and insert a `note` element after the selection.

This answer involves the selection of *an element* (and not the customary text selection) and “insert before” or “insert after” editing commands (and not the customary “insert at caret position” command). This unusual selection mode and these unusual insert commands are needed because **XXE** is in fact *an XML tree editor disguised as a word processor*.

Figure 2-3. *Tree view (displayed using menu entry "**View**|**(no style sheet)**") of the same sample TEI document*

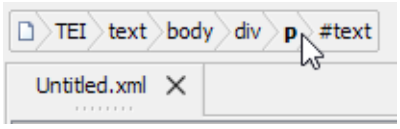


XXE has indeed:

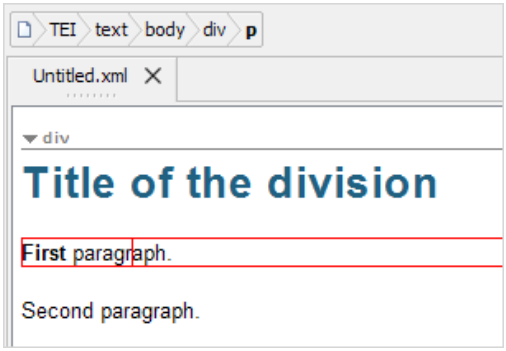
- Two selection modes: the customary text selection and a more “exotic” node selection.
- Several unusual editing commands "**Paste Before**", "**Paste After**", "**Replace**", "**Insert Before**", "**Insert After**", "**Convert**", "**Wrap**", beyond the customary **Copy**, **Cut**, **Paste**, **Delete** and **Insert** editing commands.

Now let's use the node selection and "**Insert Before**" to add a general note before first paragraph.

1. Click anywhere inside first paragraph, then click "**p**" in the node path bar.

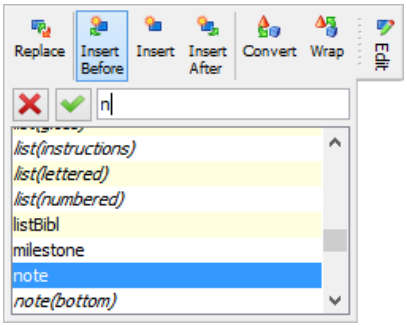
   

   First paragraph is now said to be “explicitly selected”. A red box is drawn around it.

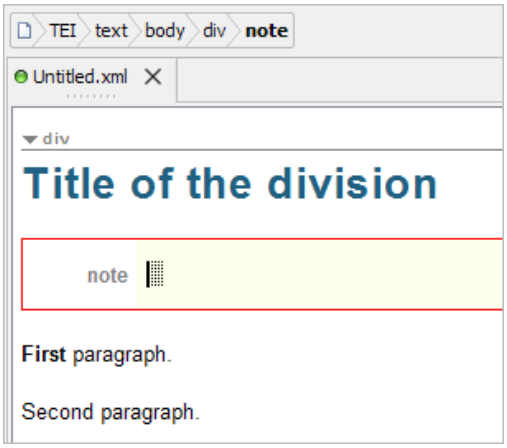


2. Click the "**Insert Before**" button in the **Edit** tool.

3. Select "**note**" from the list displayed by the **Edit** tool. The **Edit** tool supports auto-completion. Suffice to type "**n**" and then press **ENTER** to insert a `note` element.



A `note` element (explicitly selected; red box drawn around it) has been inserted before first paragraph.



You can now start typing the text of the note in the text placeholder . (The caret is already inside this text placeholder.)

Remember

Please use the tree view (displayed using menu entry "**View**|**(no style sheet)**") or the styled view with visible tags (displayed using menu entry "**View**|**Show Tags**") only when you are completely lost.

XMLmind XML Editor has been designed from the ground up to be effective to use with just the styled view (which is the default view) and the node path bar.

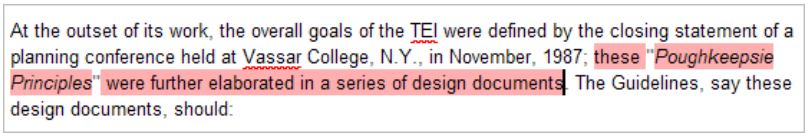
2.2.1. The selection modes

XMLmind XML Editor has two different kinds of selection, the text selection and the node selection. Moreover, the node selection may be explicit or implicit.

The text selection

The text selection works like in any word processor or text editor. However only the **Copy**, **Cut**, **Paste**, **Delete**, **Convert** and **Wrap** commands apply on the text selection. For example, you can convert the text selection to an `emph` element.

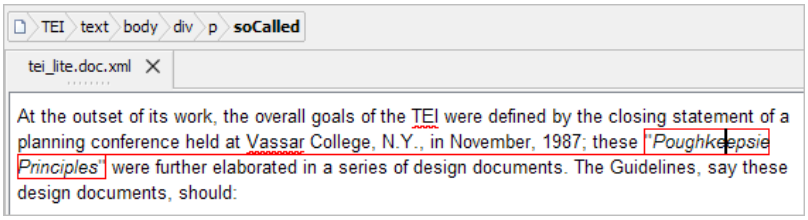
Figure 2-4. *The text selection*



The node selection

The node selection looks different than the text selection: a thin red frame is drawn around the selected nodes.

Figure 2-5. *The node selection (the `soCalled` phrase level element is explicitly selected)*



The simplest way to select a node is to click on its name in the node path bar (for example, "**soCalled**" in the above figure).

It's of course possible to select text, comment and processing instruction nodes and not only elements. The “name” of a text, comment, processing instruction node is respectively "**#text**", "**#comment**", " **#processing instruction(*TARGET*)**".

The node selection may comprise several sibling nodes, for example, two contiguous paragraphs. The easiest way to extend the node selection is to “shift-click” before or after the thin red frame.

The implicit element selection

The element directly containing the caret is always *implicitly selected*: no thin red frame around it.

For example if you want to insert a table before a paragraph, first click inside this paragraph (but not inside any of its child elements, as this would implicitly select the child element) and then use "**Insert Before**".

However, if you want to insert a list after a table, you cannot do that using the implicit element selection because a table cannot directly contain some text. You'll have to first explicitly select this table (e.g. by clicking "**table**" in the node path bar) and then use "**Insert After**".

Going further

- Online tutorial including a screencast: "*101 ways to select nodes*".

2.2.2. The basic editing commands

Toolbar buttons (or Edit menu entries)

The customary **Copy**, **Cut**, **Paste** and **Delete** commands work as expected. Of course, because **XXE** is a strictly validating XML Editor, it may forbid you to cut, paste and delete the selection depending on the editing context[3].

These other paste commands are specific to **XXE**.

Paste Before

Insert the content of the system clipboard immediately before the node selection[4].

Paste After

Insert the content of the system clipboard immediately after the node selection[4].

Edit tool buttons (or Edit menu entries)

Replace

Replace selected nodes by an element (or by a text node)[4]. For example, it may be used to replace the last element of a `div` by a new `div`, that is, a new subdivision.

Insert Before

Insert an element (or a text node) immediately before the node selection[4].

Insert

Insert an element at caret position[4].

---

[3] For example, you may not delete the `title` child element found in a `titleStmt` element (because a `titleStmt` element must start with one or more `title` elements).

[4] This command is disabled if there is a text selection.

This command may also be used to insert a child element (or a child text node) into a completely empty element. For example, it may be used to insert a `desc` child element into a `graphic` element.

Insert After

Insert an element (or a text node) immediately after the node selection[4].

Convert

When a single element is selected, **Convert** —to make it simple— changes the name of this element.

When a node range or some text is selected, **Convert** wraps the selection into a new element (just like the **Wrap** command below). For example, it may be used to convert an `l` element to a `p` element.

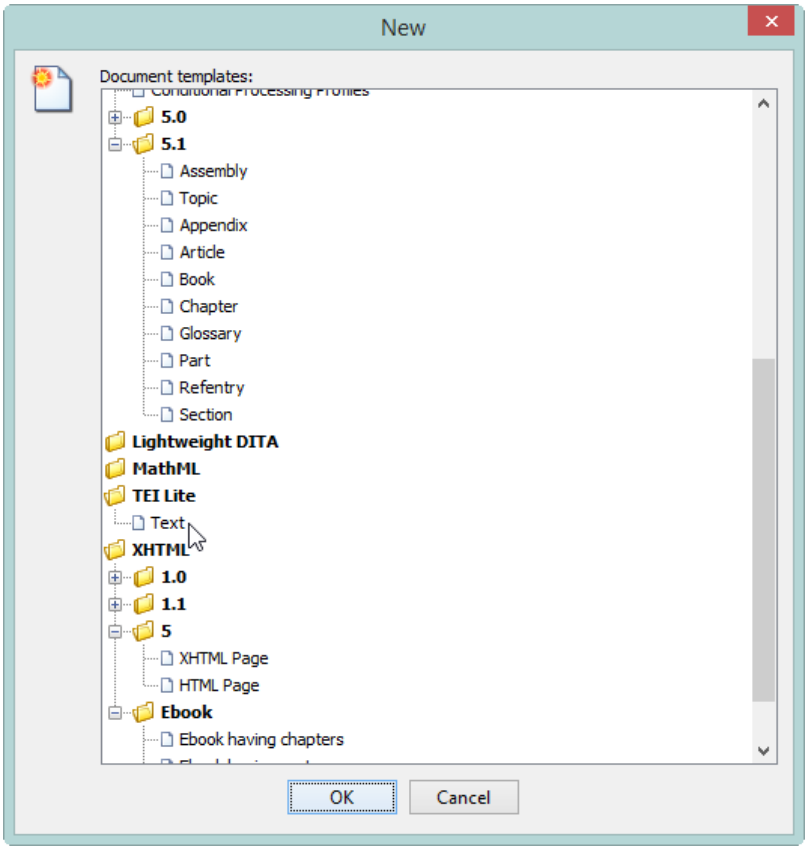
Wrap

Unlike above **Convert** command, **Wrap** always wraps the text or node selection into a new element. For example, it may be used to wrap two consecutive `p` elements into a `note` element.

3. Creating a TEI Lite document

In order to create a new TEI Lite document, click the "**New**" toolbar button [5]. This displays a dialog box:

Figure 3-1. *The "**File**|**New**" dialog box*

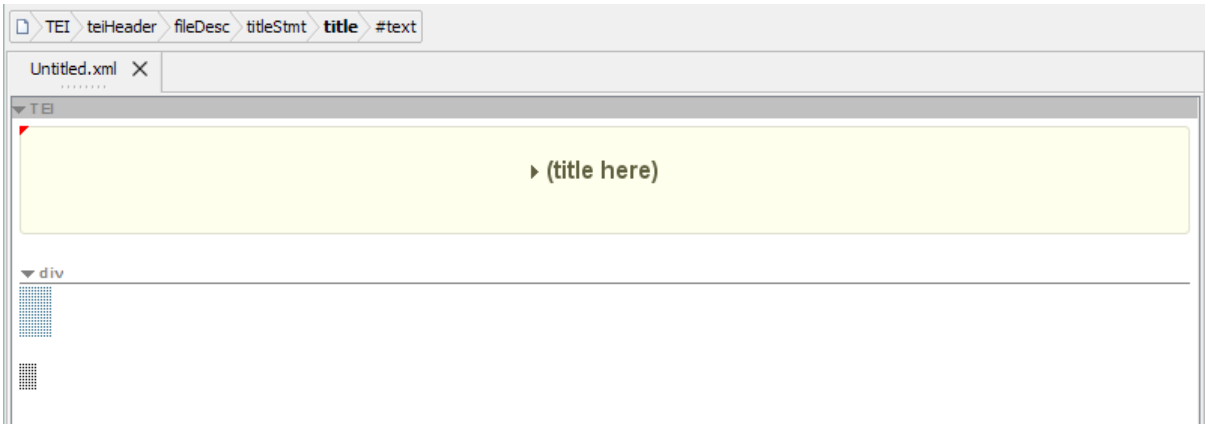


Where is my caret?

Selecting entry "**Tei Lite**|**Text**" creates an empty TEI Lite document.

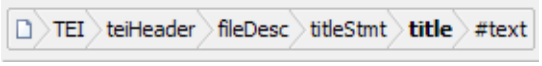
[5] Or press `Ctrl-N` (`Cmd-N` on the Mac), the keyboard shortcut of the "**File**|**New**" menu entry.

Figure 3-2. *The TEI Lite document template called "**Text**"*



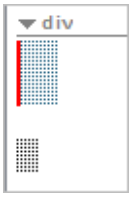
The *caret* (also called *insertion cursor*) is found deep inside in the `teiHeader` element, which is itself always initially folded[6], that's why you cannot see it. This is clearly indicated in the *node path bar* found above the document area.

Figure 3-3. *The node path bar showing that caret is currently inside a text node (`#text`), itself contained in a `title` element, itself contained in a `titleStmt` element, and so forth up to the root `TEI` element*



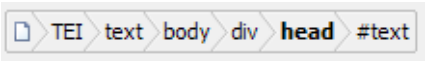
The following unusual “artifacts” are text placeholders.

Figure 3-4. *Two text placeholders, first one for the text in a `head` element and the other one for the text in a `p` element*



Click into the first, bigger, text placeholder to move the caret there. The node path bar is immediately updated.

Figure 3-5. *The node path bar showing that the caret is currently inside a `head` element*



You can now start typing the text of the `head` element of the first `div` element.

[6] Click "▸ **(title here)**" to expand the `teiHeader` element.

Going further

- Customization tip explaining how to add custom document templates to the "**File**|**New**" dialog box: "*Custom document templates*".

4. Adding a division

Click the " **Division**" toolbar button to add a new `div` element after caret position[7].

What does “add after” exactly mean?

Adding “after” indeed means adding after, but not always *immediately after*. It precisely means: after current element *or any of its parent elements, at first location where it is allowed to do so*. This last sentence has important consequences. For example:

- If you click in the `title` descendant of the `teiHeader`, this toolbar button will be grayed (disabled) because there is no way to add a `div` “after” the `title` of a `teiHeader`. Only elements like `author` (immediately after `title`), `editionStmt` (immediately after `titleStmt`), `encodingDesc` (immediately after `fileDesc`), `text` (immediately after `teiHeader`) are allowed “after” the `title` of a `teiHeader`.
- If you click in any child element of a `div` but its very last child element (e.g. if you click in the `head` of the `div`), this toolbar button will add a *sibling* `div` after current `div`.
- If you click in the very last child element of a `div`, this toolbar button will add a *child* `div`, that is, a sub-division, to current `div`.

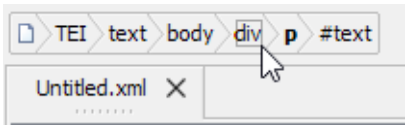
In XMLmind XML Editor, this “add after” concept is used exclusively in the toolbar buttons[8]. At first, you'll add elements at unexpected locations but after some time, you'll find it quite predictable and very handy to use.

4.1. Specifying the type of a division

You'll often want to specify the type of a newly inserted division. This division type is specified by the `type` attribute of the `div` element. Its value is "`dedication`", "`preface`", "`abstract`", etc, when the `div` is contained in `front` and "`part`", "`chapter`", "`section`", when the `div` is contained in `body`.

This is done by

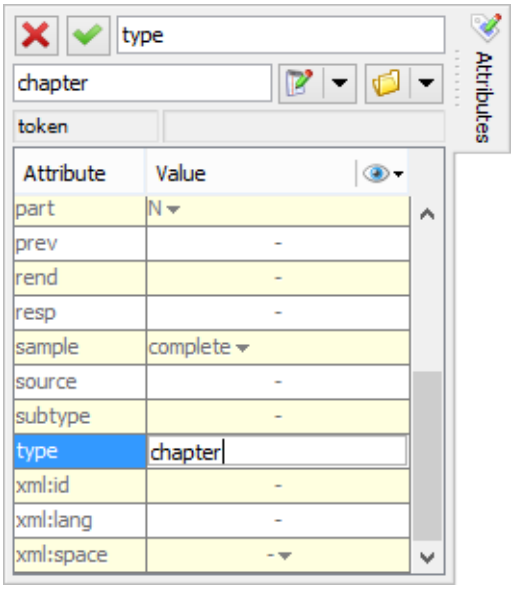
1. explicitly selecting the `div` element by clicking on "**div**" in the node path bar;



2. using the **Attributes** tool to type a value in the "**type**" row.

[7] More precisely, the new `div` is added after node selection (if any) or after caret, at a document location where it is valid to do so.

[8] Generic command " **Insert After**" inserts a new element *immediately after* implicitly or explicitly selected nodes.



Of course, the **Attributes** tool may be used to specify any attribute on any element. For example, you may also want to use it to specify the `n` or `subtype` attributes of a `div`.

Going further

- Online tutorial including a screencast: "*Quickly adding an attribute*".
- Customization tip explaining how to insert custom element templates (e.g. a `div` element having its `type` attribute already set to "`chapter`"): "*Inserting custom element templates*".

5. Paragraphs

Click the "**Paragraph**" toolbar button to add a new `p` element after caret position[9].

Handy keyboard shortcuts

The following keyboard shortcuts, commonly found in word processors, also work in XMLmind XML Editor:

- Pressing **ENTER** at the very end of a `p` element inserts a new `p` after it.
- Pressing **ENTER** at the very beginning of a `p` element inserts a new `p` before it.
- Pressing **ENTER** anywhere inside a `p` element splits this `p` in two.
- Pressing **ENTER** at the very end of a `head` element inserts a new `p` after it.

These other keyboard shortcuts are specific to **XXE**:

- Pressing **Ctrl-ENTER**[10] anywhere inside a `p` element inserts a new `p` after it.
- Pressing **Ctrl+Shift-ENTER**[10] anywhere inside a `p` element inserts a new `p` before it.

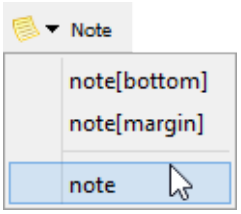
5.1. Other text blocks

Adding text blocks other than the `p` element is done using buttons also found in the toolbar.

Notes

Clicking the "**Note**" toolbar button displays a menu.

Figure 5-1. *The **Note** menu*



note[bottom]
: Insert a `note` element having attribute `place="bottom"`, that is a *footnote* (not a general note), at caret position.

note[margin]
: Insert a `note` element having attribute `place="bottom"`, that is, a *margin note*(not a general note), at caret position.

[9] More precisely, the new `p` is added after node selection (if any) or after caret, at a document location where it is valid to do so.

[10] On the Mac, press the **Cmd** key rather than the **Ctrl** key.

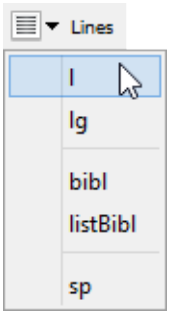
note

Add a `note` element having no `place` attribute, that is, a *general note* (e.g. a warning, a tip, etc) after caret position[11].

Lines

Clicking the "**Lines**" toolbar button displays a menu.

Figure 5-2. *The **Lines** menu*



l

Add a `l` element after caret position[11].

Note that the customary **ENTER** keyboard shortcut which works for a `p` element also works for an `l` element.

lg

Add a `lg` element after caret position[11].

bibl

Add a `bibl` element after caret position[11].

listBibl

Add a `listBibl` element after caret position[11].

sp

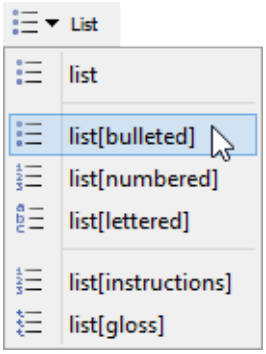
Add a `sp` element after caret position[11].

[11] More precisely, the new element is added after node selection (if any) or after caret, at a document location where it is valid to do so.

6. Lists

Clicking the " **List**" toolbar button displays a menu.

Figure 6-1. *The **List** menu*



list

Add a `list` element after caret position[12].

list[bulleted]

Add a `list` element having attribute `rend="bulleted"` after caret position[12].

list[numbered]

Add a `list` element having attribute `rend="numbered"` after caret position[12].

list[lettered]

Add a `list` element having attribute `rend="lettered"` after caret position[12].

list[instructions]

Add a `list` element having attribute `type="instructions"` after caret position[12].

list[gloss]

Add a `list` element having attribute `type="gloss"` after caret position[12].

Note

In TEI, a `p` element may have a `list` child element. However, except for a `list` element having attribute `rend="runon"` or `rend="inline"`, this element arrangement does not make much sense. That's why the above menu entries will take great care to never add a list *inside* a paragraph.

Convert a paragraph to a list

[12] More precisely, the new `list` is added after node selection (if any) or after caret, at a document location where it is valid to do so.

Another easy way to create a list is to use the ("**Increase nesting level**") toolbar button. This button converts selected `p` element to an `item`, *automatically creating the parent `list` element if needed too*. More information about this facility below.

6.1. List items

Click the "**List Item**" toolbar button to add a new list item after current list item.

Specifying current list item is easy. Suffice to click anywhere inside a list item. For `list` elements having attribute `type="gloss"`, clicking anywhere inside the `label` element preceding the `item` element works too.

In fact, this button fully supports `list` elements having attribute `type="gloss"`. For this kind of lists, it adds a `label` element followed by an `item` element.

Handy keyboard shortcuts[13]

The following keyboard shortcuts, commonly found in word processors, also work in XMLmind XML Editor:

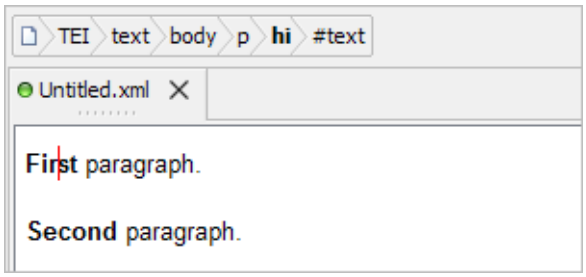
- Pressing **`ENTER`** at the very end of a `item` element inserts a new `item` after it.
- Pressing **`ENTER`** at the very beginning of a `item` element inserts a new `item` before it.
- Pressing **`ENTER`** anywhere inside a `item` element splits this `item` in two.

These other keyboard shortcuts are specific to **XXE**:

- Pressing **`Ctrl-ENTER`**[14] anywhere inside a `item` element inserts a new `item` after it.
- Pressing **`Ctrl+Shift-ENTER`**[14] anywhere inside a `item` element inserts a new `item` before it.

Converting between paragraphs and list items

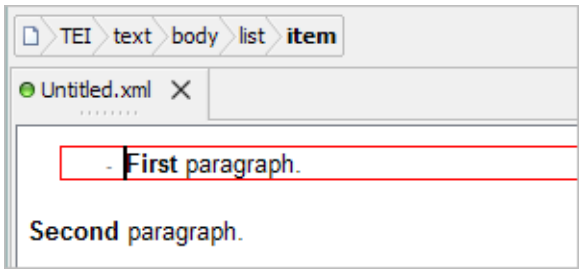
Let's see the ("**Increase nesting level**") toolbar button in action using an example. Our document contains two paragraphs.



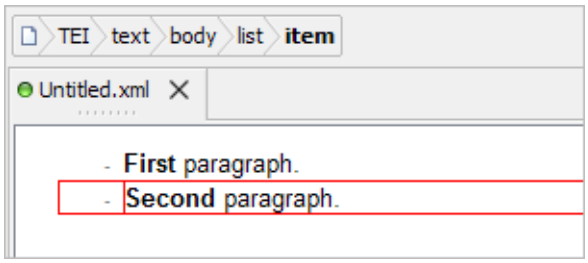
Click anywhere inside first paragraph and then click the "**Increase nesting level**" button. This converts selected `p` element to an `item` element, creating a parent `list` element on the fly.

[13] The following keyboard shortcuts will *not* work inside a `list` element having attribute `type="gloss"`.

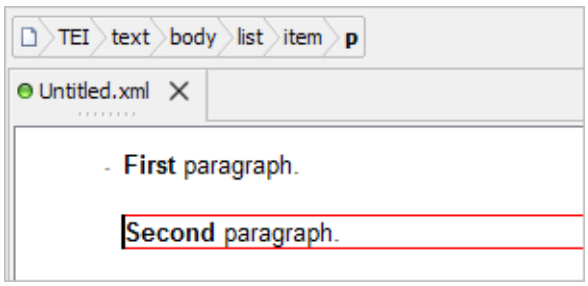
[14] On the Mac, press the **`Cmd`** key rather than the **`Ctrl`** key.



Now click anywhere inside second paragraph and then click the "**Increase nesting level**" button. This converts selected `p` element to a second `item` element, following first `item` element and sharing the same `list` parent.



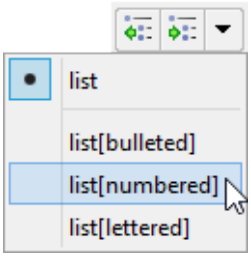
Click another time the "**Increase nesting level**" button just to see what happens. This time, selected second `item` element is converted to a `p` element and this paragraph has been made a child of first `item` element.



The ("**Decrease nesting level**") toolbar button performs the opposite operation of the ("**Increase nesting level**") toolbar button.

Tip

The ("**Increase nesting level**") toolbar button creates *plain lists* (implicitly having attribute `rend="simple"`) on the fly. However it's very easy to change the type of a list (more precisely, the value of attribute `rend`). Simply click anywhere in the list and select any of the following menu entries.



Alternatively, you can type a “prefix” to specify the kind of list to be created:

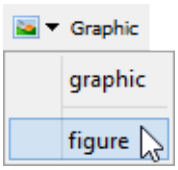
- If the paragraph is empty, simply type "`*`", "`-`", "`1.`", "`a.`", "`A.`", optionally followed by space characters, in it.
- If the paragraph is not empty, type "`*`", "`-`", "`1.`", "`a.`", "`A.`", followed by at least one space character, at the very beginning of it.

These “prefixes” are used to create respectively: **list[bulleted]**, **list[bulleted], list[numbered]**, **list[lettered]**, **list[lettered]**, as if the above "**Change list type**" menu was automatically used.

7. Figures

Clicking the "**Graphic**" toolbar button displays a menu.

Figure 7-1. *The **Graphic** menu*



graphic
: Insert a `graphic` element at caret position.

figure
: Add a `figure` element after caret position[15].

> **Note**
>
> In TEI, a `p` element may have a `figure` child element. However, this element arrangement does not make much sense. That's why the above menu entry will take great care to never add a figure *inside* a paragraph.

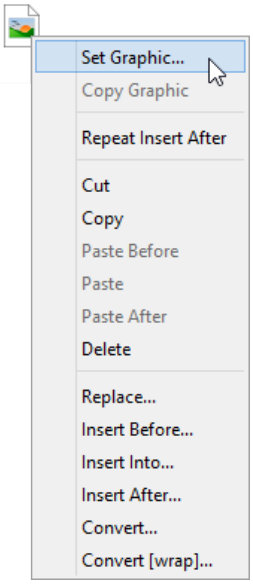
Specifying an image file

When you use any of the above menu entries, you'll be automatically prompted to specify an image file (the value of the `url` attribute of the newly inserted `graphic` element).

If, after that, you want to *change* this image file, please double-click the image or alternatively, right-click it and then select the "**Set Graphic**" entry from the contextual menu.

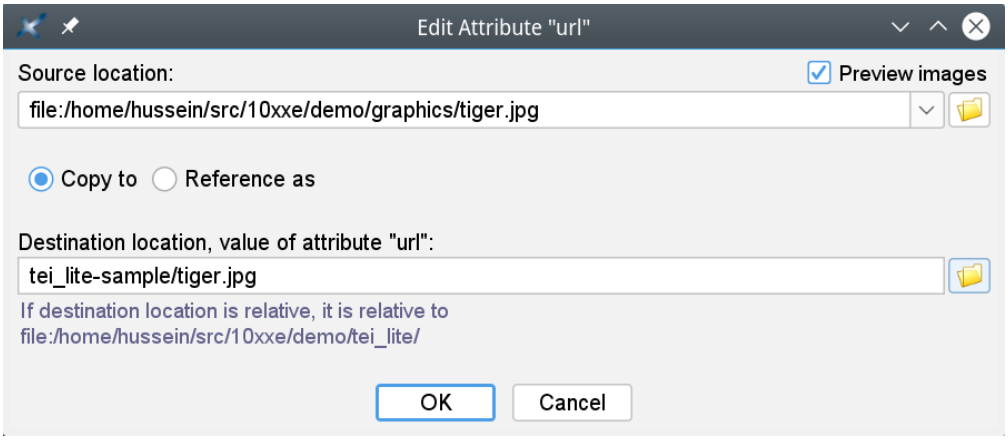
[15] More precisely, the new `figure` is added after node selection (if any) or after caret, at a document location where it is valid to do so.

Figure 7-2. *The contextual menu displayed by right-clicking an image (below it's just a placeholder because the image file has not yet been specified)*



Doing this displays a dialog box letting you specify the location of an image file. This location, generally an URL relative to the location of the document being edited, is used to set the value of attribute `url` of the `graphic` element.

Figure 7-3. *The dialog box letting you specify an image file*



- Choose an image file by clicking the **Browse** button found next to the "**Source location**" text field.

  If "**Reference as**" is selected (which is the default setting) then you are done. The location found in the "**Destination location**" text field will be used as is to set the value of attribute `url` of the `graphic` element.

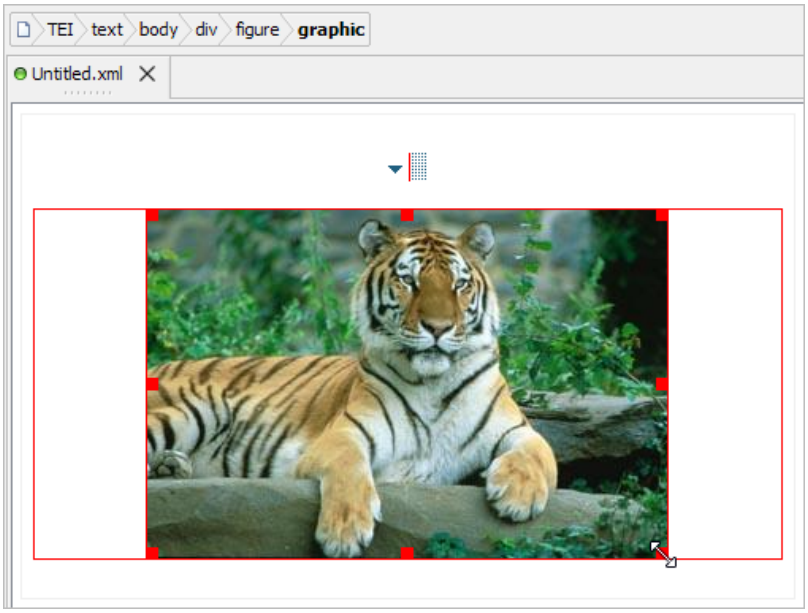
- In some cases, you'll want to *copy an image file* (i.e. not just reference it in place from your document), for example, to put it in the same folder as the one containing the document being edited.

  When this is the case, after specifying the location of the image file in the "**Source location**" text field, select "**Copy to**" and then specify in the "**Destination location**" text field where this image file is to be copied to.

For this use case too, the location found in the "**Destination location**" text field will then be used as is to set the value of attribute `url` of the `graphic` element.

Tip

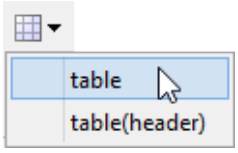
It's of course possible to resize an image by specifying the `width`, `height` or `scale` attribute of the `graphic` element in the **Attributes** tool. However the easiest way to do that is to click the image and then drag any of the red handles drawn around it.



8. Tables

Click the ("**Add table**") toolbar button to add a new `table` element after caret position[16]. This button displays a menu letting you choose between a plain table or a table starting with a header row (where first `row` child element has attribute `role="header"`).

Figure 8-1. *The menu displayed by the "**Add table**" toolbar button*

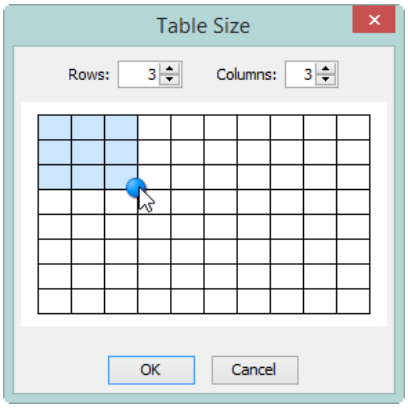


Note

In TEI, a `p` element may have a `table` child element. However, this element arrangement does not make much sense. That's why the above menu entries will take great care to never add a table *inside* a paragraph.

Selecting any of the above menu entries displays a dialog box letting you choose how many rows and columns the new table will initially have.

Figure 8-2. *The "**Table Size**" dialog box*



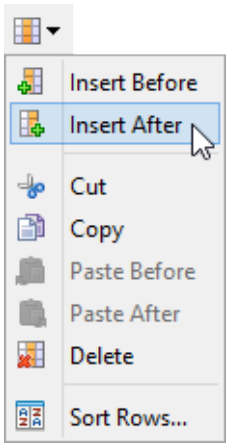
Tip

There is not need to manually set or update the `rows` and `cols` attribute of a `table` element. If needed, XMLmind XML Editor will automatically do this for you each time you'll save your document to disk.

[16] More precisely, the new `table` is added after node selection (if any) or after caret, at a document location where it is valid to do so.

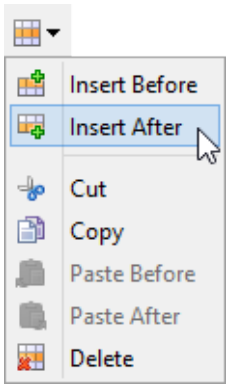
Modifying a table

The other buttons found in the "**Table**" section of the toolbar let you modify a table.

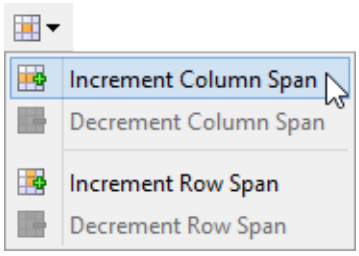


Easy column selection. Suffice to click anywhere inside a `cell` element belonging to a column to operate on this column.

> **Note**
>
> The "**Sort Rows**" menu entry displays a dialog box letting you sort the rows of the table according to the string values of the cells of current column (that is, the column to which belongs the cell containing the caret; see above easy column selection).



Easy row selection. Suffice to click anywhere inside a `cell` element belonging to a row to operate on this row.



Easy cell selection. Suffice to click anywhere inside a `cell` element to operate on this cell.

9. Phrase level elements

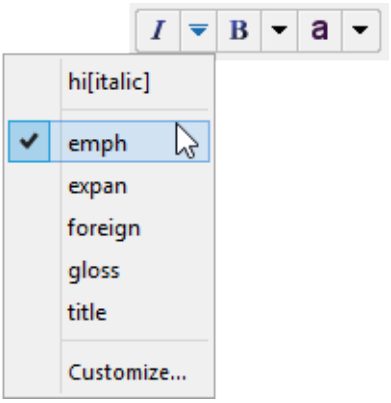
The TEI toolbar has several buttons which may be used to quickly convert text to phrase level elements (`hi`, `emph`, `abbr`, `name`, etc) and the other way round. These toolbar buttons are similar to those found in most word processors and as such, behave like “toggle switches”. The "**B**" toggle is switched on when the caret is found inside some bold text; it is off otherwise.

Figure 9-1. *The text style “toggles” found in most word processors*

A word processor has about 6 different basic text styles. TEI has dozens of different phrase level elements, so it would be impractical for the toolbar to contain dozens of toggles. Instead, the TEI toolbar has just 3 toggles (`hi` with attribute `rend="italic"`, `hi` with attribute `rend="bold"`, `name`) and 3 menus (“italic styles”, “bold styles”, “other styles”) containing entries for many more phrase level elements.

Figure 9-2. *The **hi[bold]** toggle is switched on*

Figure 9-3. *The **emph** toggle found in the “italic styles” menu is switched on*



Notice how the menu button changes to when an entry found in this menu is selected (that is, a toggle is switched on).

There are two ways to use these toggles.

- Click a toggle (e.g. “other styles” menu, **date** toggle). This creates an empty phrase level element at caret position (e.g. an empty `date` element). Start typing and when you are done, click this toggle again[17] to end the phrase level element and return to plain text.

[17] Or simply press **INS**, a generic keyboard shortcut and one of the most useful ones. More useful keyboard shortcuts in this quick reference card.

- OR make a text selection (possibly containing a mix of text and phrase level elements) then click a toggle to wrap this text selection into a phrase level element.

Remember

When there is no text selection and no node selection, all these toggles act on *the word containing the caret*. This feature is handy because it corresponds to the most common use case. However this differs from the principle of implicit element selection used everywhere else in XMLmind XML Editor.

Converting a phrase level element to plain text

If you select a phrase level element, using either the text selection or the node selection (e.g. click the phrase level element name in the node path bar), and click the corresponding toggle (which is switched on), the selection will be converted to plain text.

When there is no text selection and no node selection and a toggle is switched on, clicking this toggle converts the word containing the caret to plain text.

A handy alternative is to make a text or node selection and then click the " **Plain text**" toolbar button.

Going further

- Notice that the “italic styles”, “bold styles”, “other styles” menus all end with a "**Customize**" entry. This entry, which displays a dialog box, lets you add or remove phrase level elements to/from the menu. More information in "*Dialog box allowing to edit “text style” menu items*".

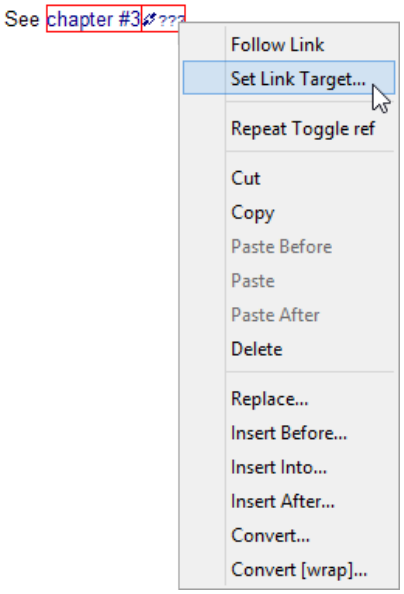
10. Links

The "**Add link**" toolbar button may be used to add a link —a `ref` or `ptr` element— to your document.

When you use any of the "**Add link**" menu entries, you'll be automatically prompted to specify the value of the `target` attribute.

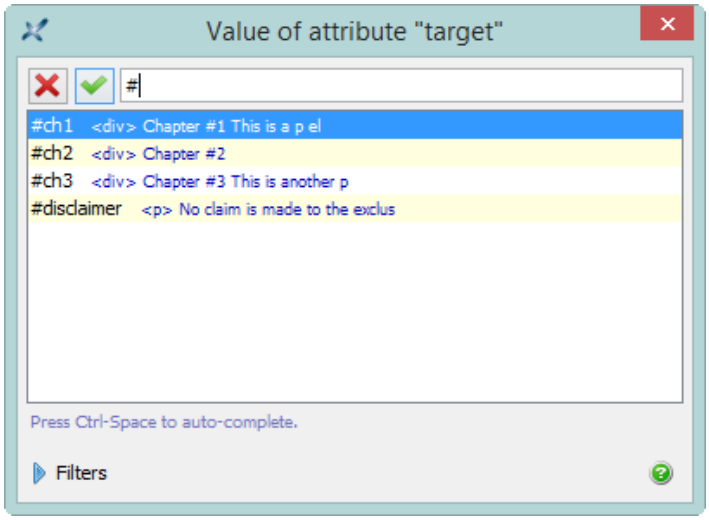
If, after that, you want to change the link target, you may of course use the **Attributes** tool just like for any other attribute. However it's more convenient to right-click the `ref` or `ptr` element and select entry "**Set Link Target**" from the contextual menu.

Figure 10-1. *The contextual menu displayed by right-clicking a link (below it's just a "???" placeholder because the link target has not yet been specified)*



This menu entry displays a dialog box implementing auto-completion. First type "#" in the text field to delete the initial "???" placeholder and thus, list all possible link targets.

Figure 10-2. *The dialog box used to set the target of a link*



An internal link uses syntax "*`#ID_OF_TARGET_ELEMENT`*" for the value of its `target` attribute. Example: "`#ch1`", where "`ch1`" is the `xml:id` of the first `div` of the document being edited. Auto-completion works great for internal links. The above dialog box automatically lists all the possible `target` values.

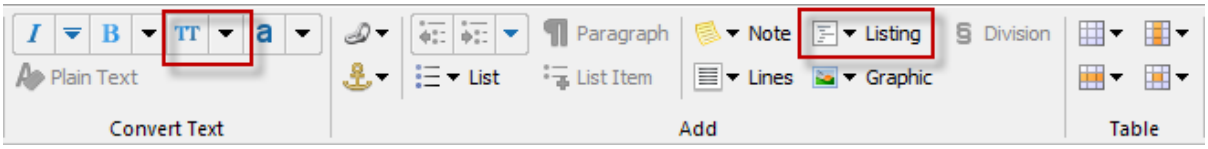
An external link typically starts with "`http://`" or "`https://`" (e.g. "`https://tei-c.org/`"). You can also use the above dialog box to type the URL of an external link, but of course, in this case, auto-completion will not be active.

11. Other features

More toolbar buttons

If you use menu entry "**Options**|**Preferences**", visit the "**General**|**Features**" section of the **Preferences** dialog box, select "**Enable the Developer Tools**" feature and restart XMLmind XML Editor, you'll get a TEI toolbar containing more buttons. The added buttons are all related to TEI elements found in technical documentation (`code`, `ident`, `gi`, `eg`, etc).

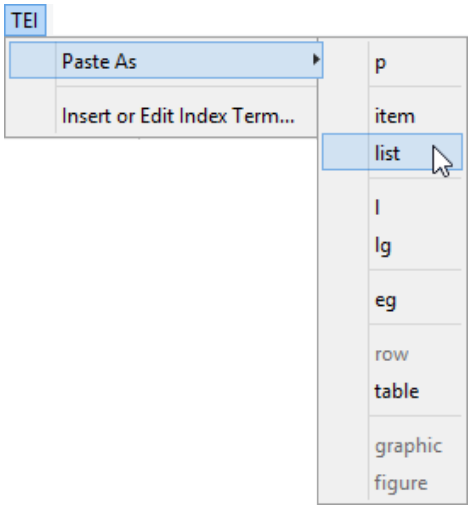
Figure 11-1. *Augmented TEI toolbar*



Paste As

The "**TEI**|**Paste As**" sub-menu lets you paste a text snippet or an image copied to the system clipboard (generally using an application other than **XXE**) as a TEI element.

Figure 11-2. *The "**TEI**|**Paste As**" sub-menu*



- Menu entry "**graphic**" replaces the text or node selection if any. When there is no selection, this menu entry pastes its element at caret position (just like **Paste**).
- All the other menu entries also replace the text or node selection if any. When there is no selection, these menu entries paste their elements at any valid position in the document following the caret position.

Replace Markdown markup by TEI elements

The "**TEI**|**Parse Markdown as XML**" menu entry (keyboard shortcut **`Ctrl+Shift+Space`**) replaces Markdown markup found all the text nodes of explicitly or implicitly selected element by equivalent TEI elements. Supported Markdown markup: italic, bold (replaced by `hi rend=bold`), bold and italic, code, strikethrough (replaced by `hi`

`rend=strikethrough`), highlight (replaced by `hi rend=underline`), subscript (replaced by `hi rend=subscript`), superscript (replaced by `hi rend=superscript`), link, image, unordered list item, ordered list item.

Example (notice the space character after "`*`"; this space character is required after all the different “block prefixes” "#", "##", "`-`", "`*`", "`1.`", etc):

```
<p>* <ident>Dart</ident> is a _programming language_.
See [Dart](https://dart.dev/).</p>
```

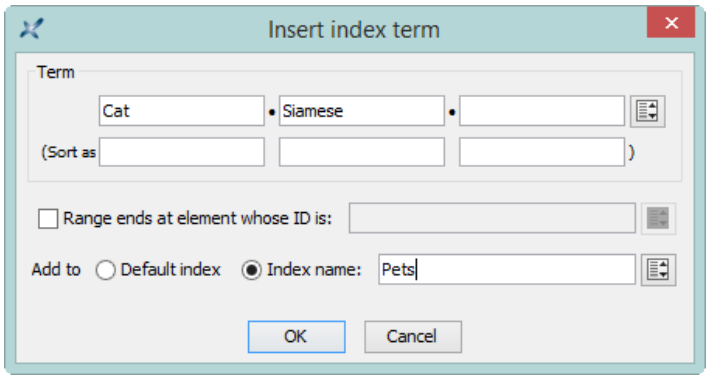
is replaced by:

```
<list rend="bulleted">
  <item><ident>Dart</ident> is a <emph>programming language</emph>.
  See <ref target="https://dart.dev/">Dart</ref>.</item>
</list>
```

Index terms

The "**TEI|Insert or Edit Index Term**" menu entry displays a dialog box which can be used to insert a new `index` element at caret position. If the caret is anywhere inside an existing `index` element, then this dialog box may be used to modify this `index` element.

Figure 11-3. *The index term editor dialog box*



Checking external links

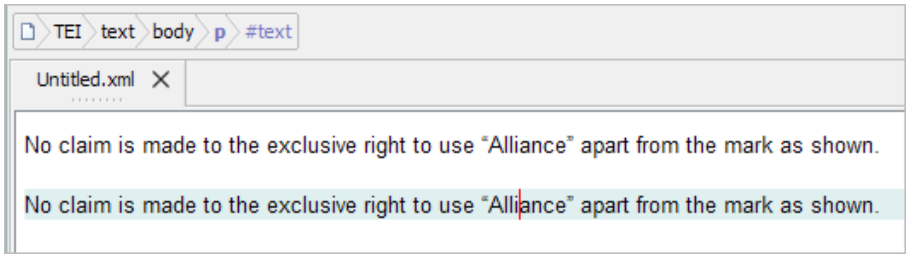
The "**Check External References**" menu entry checks all the links to external resources found in the document being edited. All kinds of external resources are checked for existence: images, audio, video, PDF documents, HTML pages, etc. When the resource is an HTML page and the link ends with a fragment (e.g. "`#bar`" in "`../doc/foo.html#bar`") then this fragment is also checked for existence.

Content reuse

XMLmind XML Editor lets you *reference* (and not only copy) in document B any element having an `xml:id` attribute found in document A[18]. Of course, document A and document B may be the same document. See figure below for example.

[18] XMLmind XML Editor fully supports the XInclude 1.1 W3C recommendation.

Figure 11-4. *A `p` element having attribute `xml:id="disclaimer"` immediately followed by a reference to itself*



An element reference cannot be modified. It is given a blue gray background indicating that the element is “read-only”. If you modify the original element and save the document to disk then all its references will be automatically updated.

Creating an element reference in **XXE** is simple and safe. It's simply a matter of using menu entry "**Edit**|**Reference**|**Copy As Reference**" and then "**Paste Before**", "**Paste**" or "**Paste After**".

Online tutorials including screencasts: "*Creating a modular document*"; "*Inserting boilerplate content*"; "*Working with master documents*".