# XMLmind XML Editor - DocBook Support

Hussein Shafie, Pixware

June 21, 2007

1. DocBook menu ... 1
1.1. Convert Document sub-menu ... 4
2. Custom bindings ... 6
3. Table rendering ... 7
3.1. HTML tables ... 8

The DocBook specific commands described below are also used by the Simplified DocBook and Slides document types.

## 1. DocBook menu

Table editing commands fully support CALS tables as well as HTML tables.

| Menu | Item | Description |
|---|---|---|
| Column<br>For a command in this menu to work, click anywhere inside a cell (or explicitly select a cell or an element having a cell ancestor). | Insert Before | Insert a column before column containing specified cell. |
| | Insert After | Insert a column after column containing specified cell. |
| | Cut | Cut to the clipboard the column containing specified cell. |
| | Copy | Copy to the clipboard the column containing specified cell. |
| | Paste Before | Paste copied or cut column before column containing specified cell. |
| | Paste After | Paste copied or cut column after column containing specified cell. |
| | Delete | Delete the column containing specified cell. |
| Row<br>For a command in this menu to work, click anywhere inside a cell (or explicitly select a cell or an element having a cell ancestor) or explicitly select a row. | Insert Before | Insert a row before row containing specified cell.<br>**Note**<br>Note that row editing commands are enabled, not only by implicitly or explicitly selecting a table cell or any of its descendants, but also by explicitly selecting a table row. |
| | Insert After | Insert a row before row containing specified cell. |
| | Cut | Cut to the clipboard the row containing specified cell. |
| | Copy | Copy to the clipboard the row containing specified cell. |
| | Paste Before | Paste copied or cut row before row containing specified cell. |
| | Paste After | Paste copied or cut row after row containing specified cell. |
| | Delete | Delete the row containing specified cell. |
| Cell<br>For a command in this menu to work, click anywhere inside a cell (or explicitly select a cell or an | Increment Column Span | Increment the number of columns spanned by specified cell. |
| | Decrement Column Span | Decrement the number of columns spanned by specified cell. |
| | Increment Row Span | Increment the number of rows spanned by specified cell. |
| | | |

| Menu | Item | Description |
|---|---|---|
| element having a cell ancestor). | Decrement Row Span | Decrement the number of rows spanned by specified cell. |

Other commands:

Convert informal<element> to/from <element>
: Converts an ``informal element" to/from a ``formal element" having a title.

  This command currently works for `informaltable`/`table`, `informalfigure`/`figure` and `informalexample`/`example`.

Go to Opposite Link End
: An element with an `ID`, `IDREF` or `IDREFS` attribute must be implicitly or explicitly selected. Scrolls to and selects opposite link end (if found). For example, will scroll to and select `<section id="foo">` if the command has been invoked with `<link linkend="foo">` selected.

Go to Next Link
: An element with an `IDREF` or `IDREFS` attribute must be implicitly or explicitly selected. Scrolls to and selects next similar link (if found). For example, will scroll to and select following `<callout arearefs="foo bar">` if the command has been invoked with `<link linkend="foo">` selected.

Go to Previous Link
: An element with an `IDREF` or `IDREFS` attribute must be implicitly or explicitly selected. Scrolls to and selects previous similar link (if found). For example, will scroll to and select preceding `<link linkend="foo">` if the command has been invoked with `<link linkend="foo">` selected.

Move Up
: Move selected element up, that is, swap it with its preceding sibling node. Requires the element to be explicitly selected.

Move Down
: Move selected element down, that is, swap it with its following sibling node. Requires the element to be explicitly selected.

Promote
: To make it simple, increase the level of selected subsection (e.g. a `sect2` element is converted to a `sect1` element).

  Requires a ``subsection" (`section`, `sect1`, `sect2`, `sect3`, `sect4` or `sect5`) or an element which is contained in the body[1] of the section to be explicitly selected.

  - If a subsection is selected, this subsection becomes a sibling of its parent section. Example: `sect2` element having `id="C"` is ``promoted":

    ```
    <sect1 id="A">...
      <sect2 id="B">...
      <sect2 id="C">...
      <sect2 id="D">...
    ```

    This results in:

    ```
    <sect1 id="A">...
      <sect2 id="B">...
    <sect1 id="C">...
      <sect2 id="D">...
    ```

[1] That is, it is not possible to ``promote" the *title* of a section.

- If another type of child element is selected, this element is wrapped in a newly created section which becomes a sibling of its parent section. Example: `para` element having `id="C"` is ``promoted":

  ```
  <sect1 id="A">...
    <para id="B">...
    <para id="C">...
    <sect2 id="D">...
  ```

  This results in:

  ```
  <sect1 id="A">...
    <para id="B">...
  <sect1>...
    <para id="C">...
    <sect2 id="D">...
  ```

Demote

To make it simple, decrease the level of selected section (e.g. a `sect1` element is converted to a `sect2` element).

Requires a ``section" (`chapter`, `appendix`, `section`, `sect1`, `sect2`, `sect3` or `sect4`) or an element which is contained in the body[2] of the section to be explicitly selected.

- If a section is selected and if this section is preceded by a section of the same type, this section becomes a subsection of its preceding sibling. Example: `sect1` element having `id="C"` is ``demoted":

  ```
  <sect1 id="A">...
    <para id="B">...
  <sect1 id="C">...
    <para id="D">...
  ```

  This results in:

  ```
  <sect1 id="A">...
    <para id="B">...
    <sect2 id="C">...
      <para id="D">...
  ```

- If a section is selected and if this section is *not* preceded by a section of the same type, a new section is created and selected section becomes a subsection of this new section. Example: `sect2` element having `id="C"` is ``demoted":

  ```
  <sect1 id="A">...
    <para id="B">...
    <sect2 id="C">...
      <para id="D">...
  ```

  This results in:

  ```
  <sect1 id="A">...
    <para id="B">...
    <sect2>...
      <sect3 id="C">...
        <para id="D">...
  ```

- If another type of child element is selected, this element and all the other ``body elements" which follow it are wrapped in a newly created subsection. Example: `para` element having `id="C"` is ``demoted":

  ```
  <sect1 id="A">...
    <para id="B">...
    <para id="C">...
    <para id="D">...
    <sect2 id="E">...
  ```

[2] That is, it is not possible to ``demote" the *title* of a section.

This results in:

```
<sect1 id="A">...
  <para id="B">...
  <sect2>...
    <para id="C">...
    <para id="D">...
  <sect2 id="E">...
```

## 1.1. Convert Document sub-menu

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

### Using the profiling stylesheets

*Profiling*, or conditional text, means that you can create a single XML document with some elements marked as conditional. More information in DocBook XSL: The Complete Guide.

If you need to use the profiling XSL stylesheets rather than the plain XSL stylesheets, use Options|Customize Configuration|Document Conversion Preferences and check option "Use the profiling stylesheets".

Note that all output formats, except Eclipse Help, have a version of the XSL stylesheets which supports profiling.

Convert to HTML..., Convert to HTML (one page)..., Convert to HTML (one page, no TOC)...

Converts the document being edited to multi page or single page HTML.

#### Generating XHTML rather than HTML

If you prefer to generate XHTML rather than HTML, use Options|Customize Configuration|Document Conversion Preferences and check option "Generate XHTML rather than HTML".

Convert to HTML Help...

Converts the document being edited to a `.chm` file. This command is disabled on platforms other than Windows.

For this command to work, the HTML Help compiler, `hhc.exe`, must have been declared as the helper application associated to files having a "`hhp`" extension. This can be specified by using the Preferences dialog box, Helper Applications section.

Convert to Java Help...

Converts the document being edited to a `.jar` file for use by the Java™ Help system.

For this command to work, the Java™ Help indexer, `jhindexer`, must have been declared as the helper application associated to files having a "`application/x-java-help-index`" MIME type. This can be specified by using the Preferences dialog box, Helper Applications section.

Convert to Eclipse Help...

Converts the document being edited to Eclipse Help.

The `eclipse.plugin.name`, `eclipse.plugin.id`, `eclipse.plugin.provider` XSL style sheet parameters must have been specified using the Options|Customize Configuration|Change Document Conversion Parameters facility.

All HTML files as well as Eclipse's `plugin.xml` and `toc.xml` are generated in the same directory:

1. This directory must be a subdirectory of the Eclipse `plugins/` directory.

2. The name of this directory must be identical to the value of the `eclipse.plugin.id` XSL style sheet parameter.

Convert to RTF..., Convert to RTF [no TOC]...

Converts the document being edited to RTF (Rich Text Format) using XMLmind FO Converter (see http://www.xmlmind.com/foconverter/). The document generated by this command can be edited and printed using Microsoft® Word 2000 and above.

Convert to WordprocessingML (MS-Word 2003)..., Convert to WordprocessingML (MS-Word 2003) [no TOC]...

Converts the document being edited to WordprocessingML using XMLmind FO Converter. The document generated by this command can be edited and printed using Microsoft® Word 2003 and above.

Convert to OpenDocument (OpenOffice.org 2)..., Convert to OpenDocument (OpenOffice.org 2) [no TOC]...

Converts the document being edited to OpenDocument (`.odt` file) using XMLmind FO Converter. The document generated by this command can be edited and printed using OpenOffice.org 2.

Convert to Office Open XML (MS-Word 2007)..., Convert to Office Open XML (MS-Word 2007) [no TOC]...

Converts the document being edited to Office Open XML (`.docx` file) using XMLmind FO Converter. The document generated by this command can be edited and printed using Microsoft® Word 2007 and above.

Print PostScript..., Print PostScript [no TOC]...

Converts the document being edited to PostScript® using RenderX XEP (see http://www.renderx.com/), if its plug-in has been installed, and Apache FOP otherwise (see http://xmlgraphics.apache.org/fop/), and then, sends the generated file to the chosen printer.

Convert to PDF..., Convert to PDF [no TOC]...

Converts the document being edited to PDF (Adobe® Portable Document Format, also known as Acrobat®) using RenderX XEP (see http://www.renderx.com/), if its plug-in has been installed, and Apache FOP otherwise (see http://xmlgraphics.apache.org/fop/).

All the above Convert commands display the URL chooser dialog box rather than the standard file chooser dialog box.

For all Convert commands except for the "Convert to HTML" command, you must specify the URL (Uniform Resource Locator) of a save file. The "Convert to HTML" command creates multiple HTML pages with a first page called `index.html`, therefore you need to specify the URL of a save directory.

Note that these commands can create directories on the fly, if needed to. For example, if you specify `http://www.acme.com/docs/report43/mydoc.html` as the URL of the save file and if directory `report43/` does not exist, this directory will be created during command execution.

### Customizing the XSL style sheets used by the above Convert commands

The XSL style sheets used to convert DocBook, Simplified DocBook and Slides document to HTML, RTF, PostScript or PDF can be customized by specifying parameters, which are name/value pairs.

The parameters of the XSL style sheets used for DocBook and Simplified DocBook are documented in "DocBook XSL Stylesheet Documentation". The parameters of the XSL style sheets used for Slides are documented in "The Slides Document Type".

The easiest way to specify your own parameters is to use menu item Options|Customize Configuration|Change Document Conversion Parameters.

An alternate method, which should be reserved to experts, consists in defining named `parameterGroup` elements in your `customize.xxe` configuration file (for example, `%APPDATA%\XMLmind\XMLEditor\addon\customize.xxe` on Windows and `$HOME/.xxe/addon/customize.xxe` on Unix). How to do this is explained in the section called “parameterGroup” in *XMLmind XML Editor - Configuration and Deployment*.

The names of the `parameterGroup`s supported by the Convert commands is found in the following tables:

| DocBook Convert command | Name of associated parameterGroup |
|---|---|
| Convert to HTML | docb.toHTML.transformParameters |
| Convert to HTML (one page) | docb.toHTML1.transformParameters |
| Convert to HTML Help | docb.toHTMLHelp.transformParameters |
| Convert to Java Help | docb.toJavaHelp.transformParameters |
| Convert to Eclipse Help | docb.toEclipseHelp.transformParameters |
| Convert to RTF, WordprocessingML, OpenDocument, Office Open XML | docb.toRTF.transformParameters |
| Print PostScript, Convert to PDF | docb.toPS.transformParameters |

| Slides Convert command | Name of associated parameterGroup |
|---|---|
| Convert to HTML | slides.toHTML.transformParameters |
| Convert to RTF, WordprocessingML, OpenDocument, Office Open XML | slides.toRTF.transformParameters |
| Print PostScript, Convert to PDF | slides.toPS.transformParameters |

### Note

It is also possible to *extensively* customize the Convert commands by specifying alternate XSLT style sheets for them. This is explained in the documentation of the `process/transform` configuration element found in the section called “Using a custom XSLT style sheet” in *XMLmind XML Editor - Commands*.

## 2. Custom bindings

| Keystroke | Action |
|---|---|
| **F2** Up | Same as menu item Move Up [2]. |
| **F2** Down | Same as menu item Move Down [2]. |
| **F2** Left | Same as menu item Promote [2]. |
| **F2** Right | Same as menu item Demote [3]. |
| **Enter** | Inside a `para/simpara`, splits the `para/simpara` in two parts. |

| Keystroke | Action |
|---|---|
| | Elsewhere, normal behaviour. |
| **Del** | At the end of a `para/simpara`, if there is no node or text selection, joins this `para/simpara` with following sibling `para/simpara` if any.<br>Otherwise, normal behaviour. |
| **BackSpace** | At the beginning of a `para/simpara`, if there is no node or text selection, joins this `para/simpara` with preceding sibling `para/simpara` if any.<br>Otherwise, normal behaviour. |
| **Ctrl-Enter** | Inside a `para`, `simpara`, `listitem`, `callout`, `step`, inserts same element after this one.<br>Elsewhere, no effect.<br>**Note**<br>If the `para/simpara` is the first child of a `listitem`, `callout` or `step`, it is a `listitem`, `callout` or `step` which is inserted, not a `para/simpara`.<br>If, in such case, you want to insert a new `para/simpara`, simply press the **Enter** key at the end of the paragraph. |
| **Shift-Ctrl-Enter** | Inside a `para`, `simpara`, `listitem`, `callout`, `step`, inserts same element before this one.<br>Elsewhere, no effect. |
| **Esc** w | Converts text copied to the clipboard using an external application (Web browser, word processor, etc) to one or more `para` elements. Then paste these paragraphs after implicitly or explicitly selected element. |

| Application Event | Action |
|---|---|
| `drop` | On an `ulink` element, changes the value of attribute `url` to the dropped string.<br>Elsewhere, considers that the dropped string is a filename or an URL and therefore, attempts to open the corresponding document in the editor. |

## 3. Table rendering

The following attributes are either completely ignored or partially supported. All other attributes are supported.

| Attribute | Support |
|---|---|
| `table` (or `informaltable`) `orient` | Ignored. |
| `table` (or `informaltable`) `pgwide` | Ignored. |
| `colspec colwidth` | All forms including "`2*`" or "`3*+1pc`" are supported. |

| Attribute | Support |
|---|---|
| | Coefficients of "*" are always converted to integers. Examples: "2.5*" is equivalent to "2*". "3.95*+0.5in" is equivalent to "4*+0.5in".<br>A column must contain at least one cell with a column span equal to 1 for the colwidth attribute to have an effect. |
| entry rotate | Ignored. |
| align | Values justify and char are rendered like left. |
| char | Ignored. See align. |
| charoff | Ignored. See align. |

## 3.1. HTML tables

DocBook supports HTML tables as well as CALS tables (that is, ``traditional" DocBook tables) starting from version 4.3. Therefore XMLmind XML Editor also supports both table models. See the section called “Table rendering” in *XMLmind XML Editor - XHTML Support* for details.

The only limitation is that mixing both HTML and CALS content models in the same table or informaltable is *absolutely not supported* by table rendering code and by table editing commands, even if this is allowed according to the DTD V4.3.

Example 1: an informaltable contains tr child elements. In such case, the informaltable is an HTML table. Setting attribute frame to topbot on this informaltable will have absolutely no visual effect.

Example 2: a table has a child tgroup element which itself contains a tbody with row/entry descendants. In such case, the table is a CALS table. Adding a thead having tr/td descendants before the tbody of the tgroup would lead to catastrophic results. Fortunately, the DocBook configuration of XMLmind XML Editor makes it hard to do this unintentionally.