XMLmind XML Editor - Creating an Ebook out of a Set of HTML Pages

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 in this document 1
    1.1. About XMLmind Ebook Compiler 1
    1.2. What you'll learn 1
2. Creating an ebook specification 3
3. Creating an ebook page 8
4. Styling an ebook 10
5. Applying conditional processing profiles to the ebooks you'll generate 11
6. Converting your ebook specification to a number of output formats 14
A. Configuring the Markdown parser 15

1. What you'll learn in this document

1.1. About XMLmind Ebook Compiler

Free, open source software XMLmind Ebook Compiler (**ebookc** for short) may be used to create a self-contained book out of a set of HTML (or Markdown) pages. If you download the distribution containing Apache FOP, all you'll need to create your first book is just a text editor.

However, **ebookc** has been tightly integrated into XMLmind XML Editor to make this task much quicker and much safer than with a text editor.

The basic idea behind **ebookc** is simple. You author a set of HTML pages and then you create an ebook specification assigning a role —part, chapter, section, appendix, etc— to each page. To make this clear, let's take a look at a very simple ebook specification:

```
<book xmlns="http://www.xmlmind.com/schema/ebook"
      href="titlepage.html">
  <frontmatter>
    <toc/>
  </frontmatter>

  <chapter href="ch1.html"/>

  <chapter href="ch2.html"/>

  <appendix href="a1.html"/>
</book>
```

Important

Before proceeding with this document, it's strongly recommended to read this short primer to quickly learn about ebook specifications and about what **ebookc** can do with these specifications.

1.2. What you'll learn

Creating an ebook out of a set of HTML pages involves the following steps:

1. Creating an ebook specification.

   This XML file, generally having a "`.ebook`" extension, contains the structure of your book. A sample ebook specification is found in *`XXE_install_dir`*`/demo/ebook-sample.ebook`.

2. Creating the ebook pages which are referenced by your ebook specification.

   An ebook page simply contains valid XHTML, preferably valid XHTML5 as **ebookc** anyway generates XHTML5 markup.

Tip

In addition to HTML, an ebook page may also be written in Markdown. More information in Markdown support.

See also Appendix A. Configuring the Markdown parser.

3. Creating a CSS stylesheet which will make your ebook look good.

   Note that this CSS stylesheet also applies to output formats like PDF or DOCX which, unlike EPUB and Web Help, are not based on XHTML and CSS.

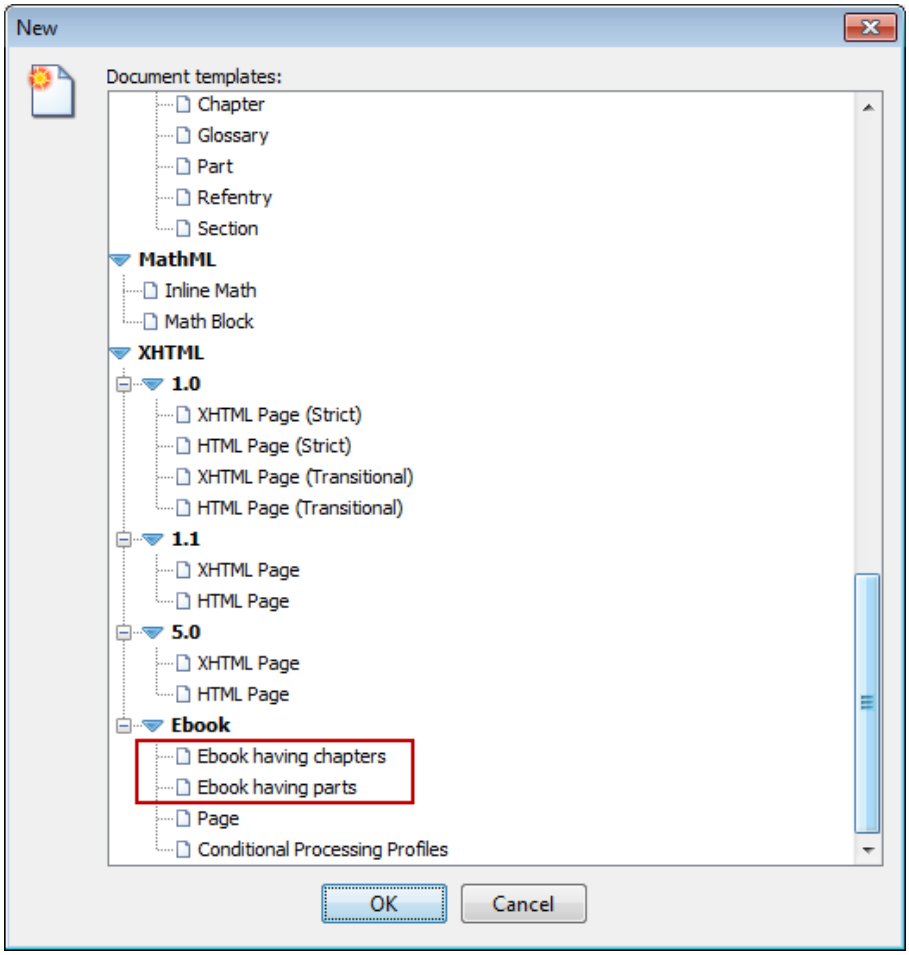
4. Optionally creating one or more conditional processing profiles.

   This step is often needed when the content of the ebook depends on the target format. For example, some ebook pages may contain interactive content (e.g. a feedback form) and this interactive content simply cannot be rendered in PDF or DOCX.

5. Converting your ebook specification to a number of output formats: EPUB, Web Help, PDF, RTF, WML, DOCX (MS-Word) and ODT (OpenOffice/LibreOffice).

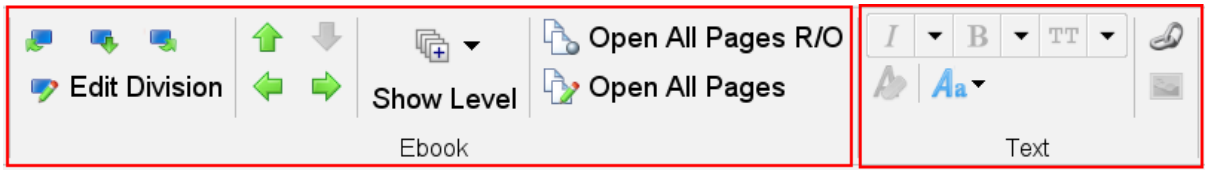
2. Creating an ebook specification

An ebook specification is an XML generally having a "`.ebook`" file name extension. This file may be created by first selecting **File > New** and then choosing one of the two ebook specification templates.



An newly created ebook specification is then rendered as follows:

Notice that the ebook specification toolbar is divided in two parts.



- The first part of the toolbar contains buttons which may be used to edit the structure of the book.
- The second part of the toolbar is active only inside element `head/title`. It contains the same buttons as those found in an XHTML page toolbar and may be used to edit the “rich text” possibly contained in a `title`.

The first part of the toolbar contains the following buttons:

Insert Book Division Before

Insert a book division (chapter, section, appendix, etc) before selected book division. Displays the "**Insert or Edit Book Division**" dialog box. Keyboard shortcut: **`Shift-ENTER`**.

Insert Book Division Into

Insert a book division (chapter, section, appendix, etc) as the last child of selected book division. Displays the "**Insert or Edit Book Division**" dialog box. Keyboard shortcut: **`Ctrl-ENTER`**.

Insert Book Division After

Insert a book division (chapter, section, appendix, etc) after selected book division. Displays the "**Insert or Edit Book Division**" dialog box. Keyboard shortcut: `ENTER`.

Edit Book Division

Displays the "**Insert or Edit Book Division**" dialog box in order to modify selected book division (chapter, section, appendix, etc). Keyboard shortcut: `ESC e`.

Move Up

Move selected element up, that is, swap it with its preceding sibling node. Requires the element to be explicitly selected. Keyboard shortcut: `Alt+Shift-UP`.

Move Down

Move selected element down, that is, swap it with its following sibling node. Requires the element to be explicitly selected. Keyboard shortcut: `Alt+Shift-DOWN`.

Promote

Decrease the nesting level of selected book division (chapter, section, appendix, etc). For example, convert a `section` to a `chapter`. Keyboard shortcut: `Alt+Shift-LEFT`.

Demote

Increase the nesting level of selected book division (chapter, section, appendix, etc). For example, convert a `chapter` to a `section`. Keyboard shortcut: `Alt+Shift-RIGHT`.

Show Level

Displays a menu containing "**Show Level 1**", "**Show Level 2**", ..., "**Show Level 9**" items. "**Show Level *N***" means: expand all the collapsible elements of the ebook specification up to nesting level *N* and recursively collapse all the collapsible elements having a nesting level greater than *N*.

Open All Pages R/O

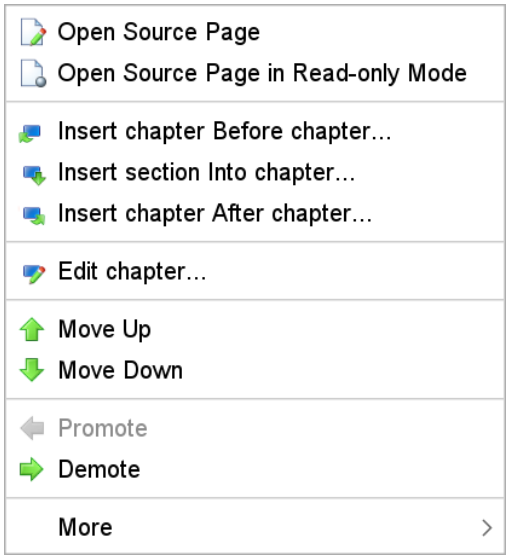
Opens in read-only mode all the HTML source pages referenced in the selected elements and their descendants.

Open All Pages

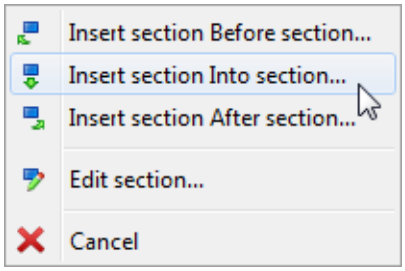
Opens in normal read-write mode all the HTML source pages referenced in the selected elements and their descendants.

Tip

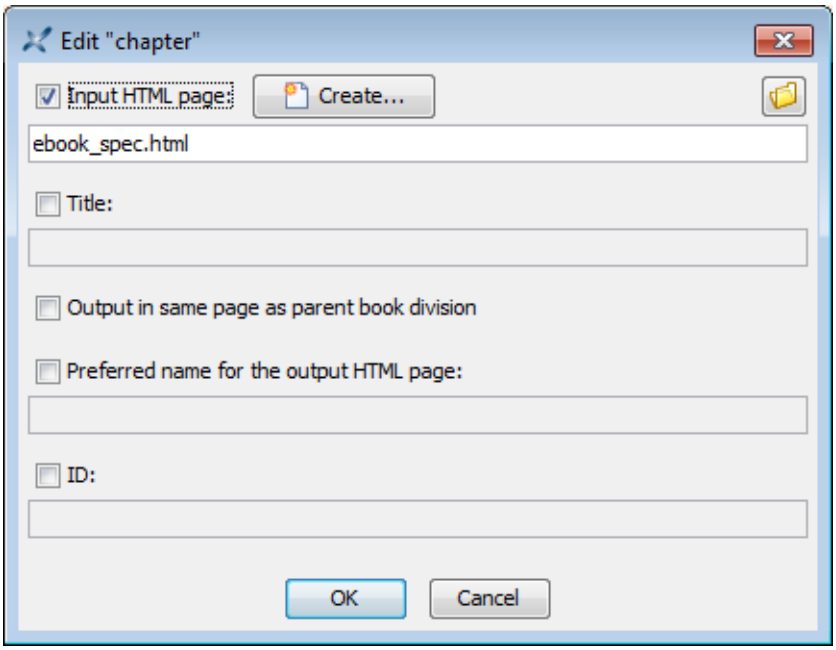
Right-clicking (`Ctrl-click` on the Mac) a book division (chapter, section, appendix, etc) displays a contextual menu containing some of the buttons found the ebook specification toolbar.



Dropping a file or URL onto a book division (chapter, section, appendix, etc) displays a contextual menu containing some of the buttons found the ebook specification toolbar.



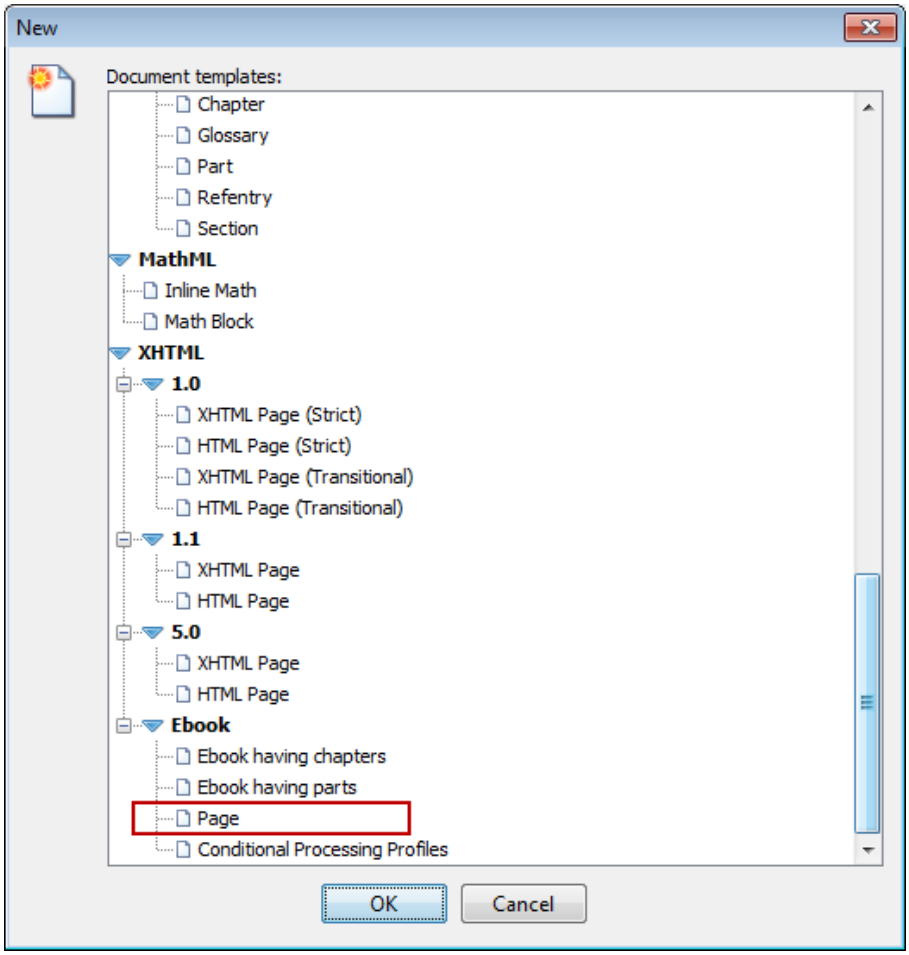
The "Insert or Edit Book Division" dialog box



This dialog box displayed by the **Insert Book Division Before**, **Insert Book Division Into**, **Insert Book Division After** and **Edit Book Division** tool bar buttons let's the user easily add or remove the main attributes of a book division (chapter, section, appendix, etc). Note that the `book` root element is also considered to be a book division.

3. Creating an ebook page

An ebook page is simply an XHTML5 file where the root `html` element has attribute `class="role-ebook-page"`. This file can be created by first selecting **File** > **New** and then choosing the "**Page**" template.



Important

Attribute `class="role-ebook-page"` is not required to use an HTML page in an ebook specification. Simply XMLmind XML Editor automatically detects this attribute and replaces its stock "**XHTML**" menus and toolbars by extended "**XHTML**" menus and toolbars.

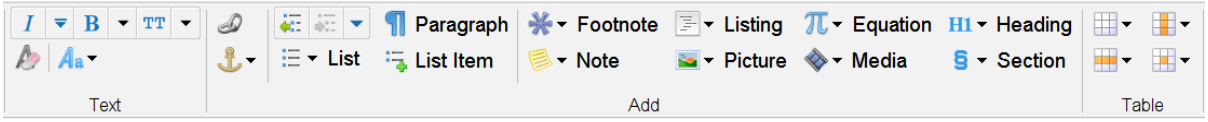
Moreover you are not restricted to using pages containing XHTML5 markup. You may use pages containing XHTML 1.0 Strict or Transitional or XHTML 1.1 if you prefer. However you must keep in mind that **ebookc** anyway generates XHTML5 markup. In all cases, you have to make sure that the contents of your input pages is valid.

Tip

In addition to HTML, an ebook page may also be written in Markdown. More information in Markdown support.

See also Appendix A. Configuring the Markdown parser.

The "**XHTML**" menus and toolbars displayed for an ebook page slightly differ from those displayed when opening a plain XHTML5 document.



The toolbar contains the following specific buttons:

Add footnote

Menu entry "**span(footnote)**" inserts a element at caret position or converts selected text to this element. Menu entry "**div(footnote)**" adds a general footnote, that is, both a footnote body (element `<div class="role-footnote" id="ID">`) and the corresponding footnote marker (element `<a class="role-footnote-ref" href="#ID">`). See "*Content of a source HTML page, Footnotes*".

Note that in order to delete a general footnote suffice to delete its body or its marker; no need to delete both. This will be done automatically the next time you'll save your ebook page to disk.

Add note

These menu entries add admonitions like the `<blockquote class="role-important">` element. See "*Content of a source HTML page, Admonitions*".

Add pre

Menu entry "**pre(listing)**" adds a <pre class="role-listing-1"> element. Menu entry "**figure(example)**" adds a `<figure class="role-example">` element. See "*Content of a source HTML page, Examples*".

Add MathML equation

Menu entry "**figure(equation)**" adds a `<figure class="role-equation">` element. See "*Content of a source HTML page, Equations*".

4. Styling an ebook

The ebook you'll generate using submenu **Ebook > Convert Document** is styled using CSS stylesheets.

Adding a good CSS stylesheet to your ebook is really a required step. Without this CSS stylesheet, the ebook you'll generate would be *very poorly formatted*.

Two goods news though:

- Setting attribute `includebasestylesheet` of element `book` to "`true`" guarantees to effortlessly create a nicely formatted book. More information about this attribute in *Leveraging `base.css`, the stock CSS stylesheet*.

Note

Alternatively, you may want to use a custom CSS stylesheet developed from scratch, starting from the template found in *`ebookc_install_dir`*`/doc/manual/template/skeleton.css`. In this case, as explained in *Nicely formatted books*, make sure to add this kind of link to the `headcommon` element of your root `book` element:

```
<headcommon>
  <html:link href="my_custom_stylesheet.css"
             rel="stylesheet" type="text/css"/>
</headcommon>
```

- The CSS styles specified in the ebook specification and in the source HTML pages are also used when generating output formats like PDF, RTF, DOCX, even if these formats are not directly related to HTML and CSS.

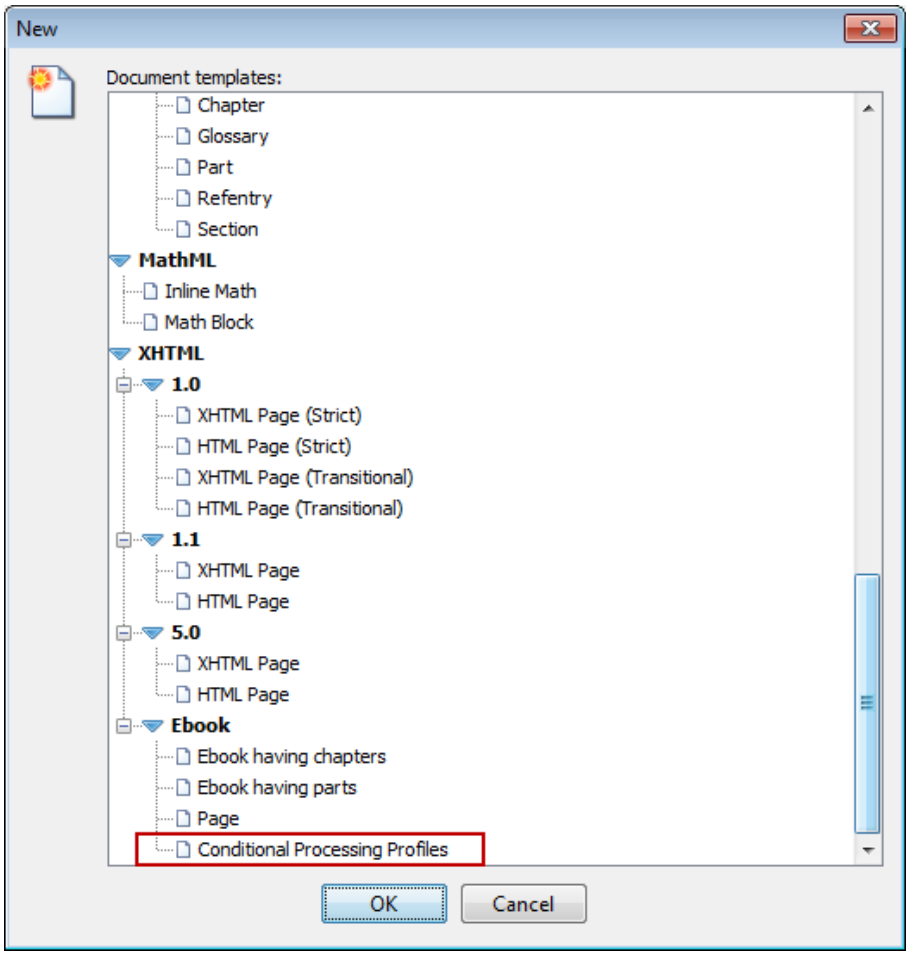
5. Applying conditional processing profiles to the ebooks you'll generate

What is conditional processing is explained in "*Handy features, Conditional processing*".

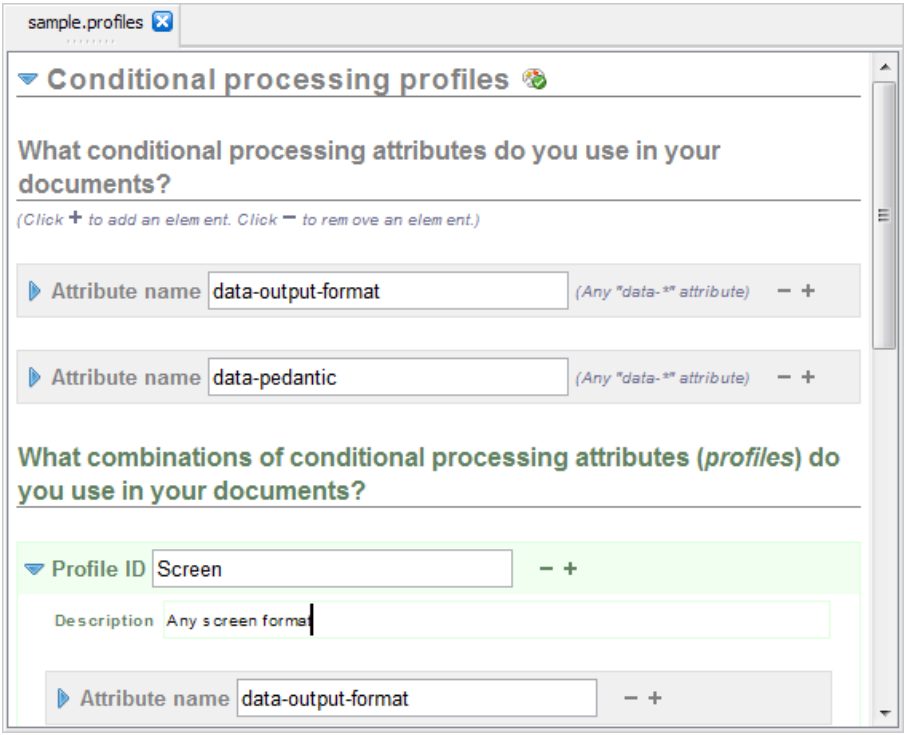
With XMLmind XML Editor, the only way to apply conditional processing profiles to the ebooks you'll generate is to use the facilities of the "Easy Profiling" add-on (installed by default).

Applying conditional processing profiles to the ebooks you'll generate involves the following steps:

1. Create a "`.profiles`" file. This is done by first selecting **File > New** and then choosing the "**Conditional Processing Profiles**" template.



2. Specify the "`data-*`" attributes you'll use for profiling and the values allowed for each of these attributes. Specify one or more of conditional processing profiles.

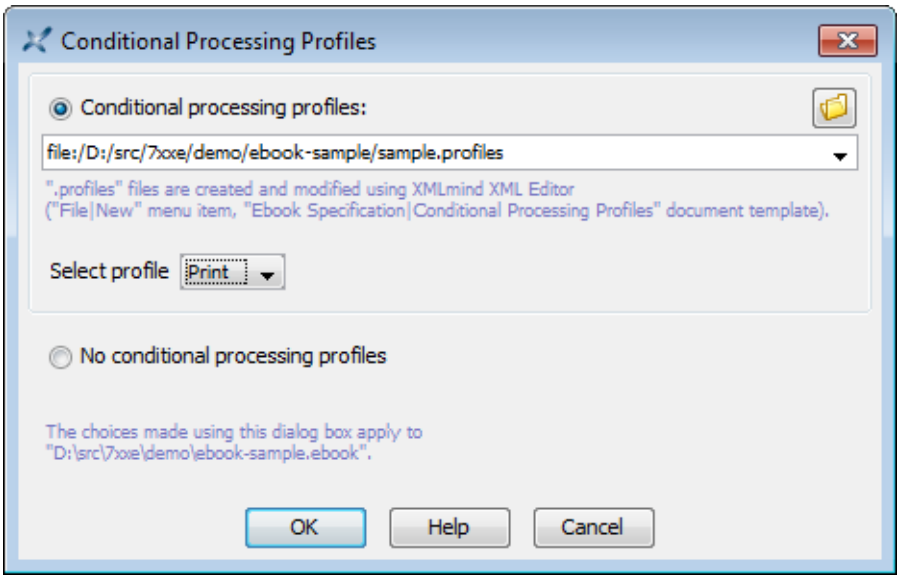


The above screenshot shows *`XXE_install_dir`*`/demo/ebook-sample/sample.profiles` which is part of the *`XXE_install_dir`*`/demo/ebook-sample.ebook` demo.

- This "`.profiles`" file specifies that 2 profiling attributes may be used in the `ebook-sample.ebook` demo: `data-output-format` and `data-pedantic`.
- It defines 3 conditional processing profiles called "Screen", "Print" and "Online".
- It also gives distinctive screen styles to elements belonging to any of the unselected conditional processing profile (e.g. in **XXE**'s styled view, strike through excluded `span`s), which is a very convenient feature of the "Easy Profiling" add-on.

3. Turn on **View > Use as Master Document** for the ebook specification if this is not already the case. More information in "*XMLmind XML Editor - Easy Profiling, Associating the ".profiles" file to the map rather than to individual topics*".

4. Associate a conditional processing profile to the ebook specification using **Ebook > Conditional Processing > Select Profile**.



5. Convert the the ebook specification using **Ebook > Convert Document**.

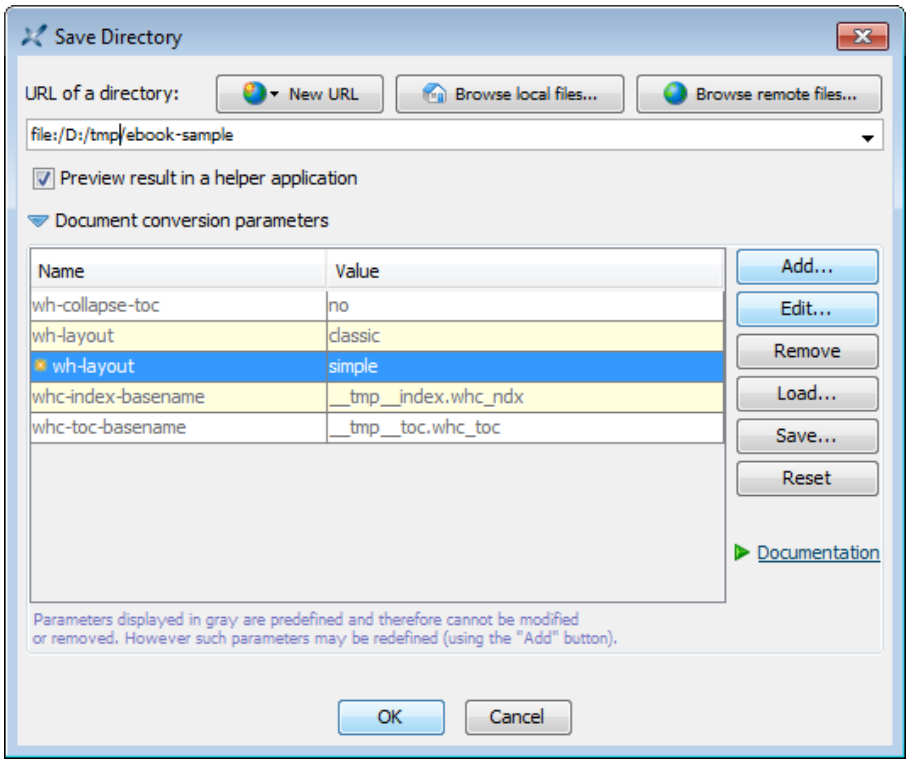
6. Repeat steps 5 and 6 with different conditional processing profiles and/or different output formats.

6. Converting your ebook specification to a number of output formats

Converting your ebook specification to a number of output formats is done by using the entries of submenu **Ebook > Convert Document**. However, before attempting to use any of these menu entries

1. Make sure to save your ebook specification to disk.
2. You may want to check the cross-references of the ebook you'll generate using **Ebook > Check Ebook**.

The content of the ebook you'll generate may be customized the usual ways, for example by specifying a number of XSLT stylesheet parameters. These XSLT stylesheet parameters are all documented in "*XMLmind Ebook Compiler Manual, XSLT stylesheets parameters*".



In the above screenshot (see *The "Document conversion parameters" panel*), the user specifies that she/he prefers a simple layout rather than the classic layout for the Web Help she/he is generating.

However it's also possible to customize the content of the ebook you'll generate by setting a number of attributes of the root `book` element of your ebook specification. For example, `parttocdepth="100"` instructs **ebookc** to automatically generate a Table of Contents at the beginning of each part of the book.

A. Configuring the Markdown parser

There are many “flavors” of Markdown. The flavor you use to write some of the ebook pages may differ from what is supported out of the box by the Markdown parser which is part of XMLmind Ebook Compiler. Fortunately, this Markdown parser[1] is highly configurable.

The Markdown parser may be configured by defining a system property called "`ebookc.load.options`" containing one or more `load.markdown.`*XXX* options. These options are all documented in Markdown support.

The "`ebookc.load.options`" system property is best defined in a `customize.xxe` configuration file. Example:

```
<property name="ebookc.load.options">
  load.markdown.renderer.RENDER_HEADER_ID true

  load.markdown.emoji true
  load.markdown.emoji.ATTR_IMAGE_SIZE 16
  load.markdown.emoji.ATTR_ALIGN ""
  load.markdown.emoji.USE_IMAGE_TYPE IMAGE_ONLY
  load.markdown.emoji.USE_SHORTCUT_TYPE ANY_GITHUB_PREFERRED
</property>
```

More information about the `property` configuration element and the `customize.xxe` configurations files in *XMLmind XML Editor - Configuration and Deployment*.

[1] Implemented using the flexmark-java software component.