XMLmind XML Editor - Easy Profiling

Hussein Shafie, XMLmind Software <xmleditor-support@xmlmind.com>

June 30, 2026

Abstract

The "**Easy Profiling**" add-on makes it convenient and easy working with conditional processing profiles in XMLmind XML Editor (**XXE** for short). This document explains how to use the facilities added by this add-on to **XXE**.

Table of Contents

1. What is the "**Easy Profiling**" add-on? 1
2. Working with conditional processing profiles *without* the facilities added by the "**Easy Profiling**" add-on 2
3. Using the facilities added by the "**Easy Profiling**" add-on 3
    3.1. Associating the "`.profiles`" file to the map rather than to individual topics 8
4. About "`.profiles`" files 9
    4.1. How it works 9
    4.2. Creating a "`.profiles`" file allowing to flag contents in the deliverables 9
5. Dialog box reference 11
    5.1. The "**Set Style**" dialog box 11
    5.2. The "**Select Profile**" dialog box 13
    5.3. The "**Set Profiling Attributes**" dialog box 14
A. Integrating the "**Easy Profiling**" add-on with document types other than DITA, DocBook and Ebook 15
    1. The `profiling` configuration element 18
    2. Profiling commands 19
        2.1. Command `setProfiling` 19
        2.2. Command `setProfilingAttributes` 19

1. What is the "Easy Profiling" add-on?

> If you are interested in working with conditional processing profiles, you need to make sure that the "**Easy Profiling**" add-on has been installed. This is done using menu item "**Options** → **Install Add-ons**".

The "**Easy Profiling**" add-on adds a submenu called "**Conditional Processing**", document templates called "**Conditional Processing Profiles**" to the DITA, DocBook and Ebook configurations and an "**Add-on**|**Easy Profiling**" preferences sheet to the **Preferences** dialog box.

A "**Conditional Processing Profiles**" document template may be used to create a "`.profiles`" file in XMLmind XML Editor. Such files contain a specification of one or more conditional processing profiles in a proprietary format.

The "**Conditional Processing**" submenu lets the user work with conditional processing profiles when the document being edited is a DITA map or topic (of any kind), a DocBook 4 or 5 document, an ebook specification or an ebook source XHTML5 page.

This add-on is incompatible with the use of some advanced DITA 1.3 features

If your DITA maps or topics use:

- branch filtering (e.g. `ditavalref`)
- AND/OR grouped values for conditional processing attributes (e.g. `product="a database(dbA dbB) b appserver(mySERVER)"`)

then it is strongly recommended *not to install and use* the "**Easy Profiling**" add-on. Anyway, this add-on is not meant to be used by DITA experts who master the two aforementioned advanced features.

A tutorial on the "**Easy Profiling**" add-on, including a screencast, is found in *Conditional processing made easy*.

2. Working with conditional processing profiles *without* the facilities added by the "Easy Profiling" add-on

Conditional processing, also called *profiling* or conditional text, means that you can create a single XML document with some elements marked as conditional. When you process such a document, you can specify which conditions apply for that version of the output, and the XSLT stylesheet or the DITA processor will include or exclude the marked text to satisfy the conditions.

Let's suppose you are writing a manual for a software product which is available in 3 editions: Lite, Professional and Ultimate. You want to author a single manual (as the vast majority of its contents is common to the 3 editions), but you want to generate 3 different PDF deliverables, one for each edition (otherwise your customers would find your manual mixing info about the 3 editions confusing to read).

DITA example: the conditional processing attribute used in the example below is `product`. It's a standard DITA profiling attribute.

```
<p>First make sure that your work has been saved to disk.</p>

<p product="Lite">You need to install a third-party application such as
<xref format="html" href="http://www.pdfforge.org/pdfcreator"
scope="external">PDFCreator</xref>. Once this is done, select menu item
<menucascade> <uicontrol>File</uicontrol><uicontrol>Print</uicontrol>
</menucascade> and choose to print to the “PDF printer” added by this
third-party application.</p>

<p product="Professional Ultimate">Then simply select menu item
<menucascade><uicontrol>File</uicontrol><uicontrol>Export as PDF</uicontrol>
</menucascade>.</p>

<p product="Professional">Note that dialog box displayed by this menu item
has no "<uicontrol>PDF/A Format</uicontrol>" checkbox. This format is
available only in the Ultimate Edition.</p>

<p product="Ultimate">The dialog box displayed by this menu item has a
"<uicontrol>PDF/A Format</uicontrol>" checkbox. Check "<uicontrol>PDF/A
```

```
Format</uicontrol>" if you want to create a PDF file well-suited to
long-term archiving.</p>
```

First paragraph having no `product` attribute applies to all editions and thus, will be included in all deliverables. Second paragraph applies only to the Lite edition. Third paragraph applies to both Professional and Ultimate editions. Fourth paragraph applies only to the Professional edition. Fifth paragraph applies only to the Ultimate edition.

For a DITA document, a conditional processing profile, that is, a specification of which marked text should be included, excluded or flagged, is expressed using a DITAVAL file[1].

The following "`.ditaval`" file may be used to generate the deliverable for the Lite edition:

```
<val>
  <prop action="exclude" att="product" />
  <prop action="include" att="product" val="Lite" />
</val>
```

You must pass the location of this "`.ditaval`" file as a parameter to the DITA processor in order to generate the deliverable for the Lite edition. This deliverable will contain only the first and second paragraphs.

Working with conditional processing profiles without any assistance poses the following problems:

1. Setting conditional processing attributes on some elements is tedious and error prone. For example, it's common to make a typo in the value of the attribute ("`light`" instead of "`Lite`").
2. There is no way to visualize on screen which are the marked elements other than switching to the source view of the document.
3. The "`.ditaval`" files must be created by hand. In the case of the above DITA example, you'll have to create 3 "`.ditaval`" files, one for each profile, which is tedious and error prone.
4. You must explicitly pass the location of the "`.ditaval`" file as a parameter to the DITA processor when you generate a deliverable. This too can quickly become tedious and error prone.

3. Using the facilities added by the "Easy Profiling" add-on

Important preliminary step if you author DocBook documents

If you author DocBook documents and want to work with conditional processing profiles, then please use "**Options → Customize Configuration → Customize Document Conversion Stylesheets**" to select the *profiling* XSLT stylesheets. This step is needed because, out of the box, it's the normal XSLT stylesheets, which do not support conditional processing, which are selected by default.

Note that there is no such preliminary step if you author DITA documents.

Using the facilities added by the "**Easy Profiling**" add-on involves the following steps:

[1] For a DocBook document, which marked text should be included or excluded is expressed using special XSLT stylesheet parameters. More information in DocBook XSL: The Complete Guide.

1. Create a "`.profiles`" file, specifying a set of conditional processing profiles, in XMLmind XML Editor.
2. Associate the "`.profiles`" file to the document being edited.
3. Set profiling attributes on some elements contained in the document being edited.
4. Select one of the available profiles.
5. Generate the corresponding deliverable.

We'll use here the same DITA example as in Section 2, “Working with conditional processing profiles *without* the facilities added by the "**Easy Profiling**" add-on” [2]. All the XML files used in this example are found in directory `sample/`.

1. Create `editions.profiles` in XMLmind XML Editor.

   This is done using **File** → **New** and then selecting the **DITA**|**Conditional Processing Profiles** document template from the dialog box.

   a. In the first section titled "**What condition processing attributes...**", specify that you want to use standard profiling attribute `product` and that you allow it to have the following values: `Lite`, `Professional`, `Ultimate`.

      The ▾ menu which is next to the "**Attribute name**" field lets the user choose the name of a standard profiling attribute.

   b. In the second section titled "**What combinations...**", define profile "`lite`" as having attribute `product` with value `Lite`.

      The ▾ menu which is next to the "**Attribute name**" field lets the user choose the name of one of the attributes specified in section "**What condition processing attributes...**".

      The ▾ menu which is next to the "**Has value**" field lets the user choose one of the allowed attribute values.

   c. Repeat the previous step to define profile "`pro`" and to define profile "`ultimate`".

   d. It's possible to automatically add some styles to elements having a given profiling attribute value. This facility makes it easy spotting on screen which are the elements which have been marked for conditional processing.

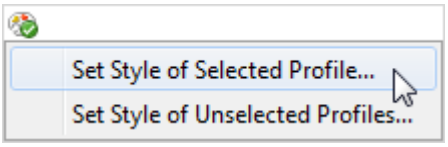
      This is done by clicking the button which is next to each "**Allow value**" field. This button displays the "**Set Style**" dialog box [11].

      Use this dialog to give a dark red text color to all elements having attribute `product` containing value `Lite`.

   e. Repeat the previous step to give a dark green color to all elements having attribute `product` containing value `Professional` and a dark blue color to all elements having attribute `product` containing value `Ultimate`.

   f. Besides styling individual profiling attributes, it's also possible to style the elements which are going to be included or excluded from the deliverables (PDF, RTF, HTML, etc).

      This is done by clicking the button which is next to the "**Conditional processing profiles**" document title. This button displays a menu having two entries:



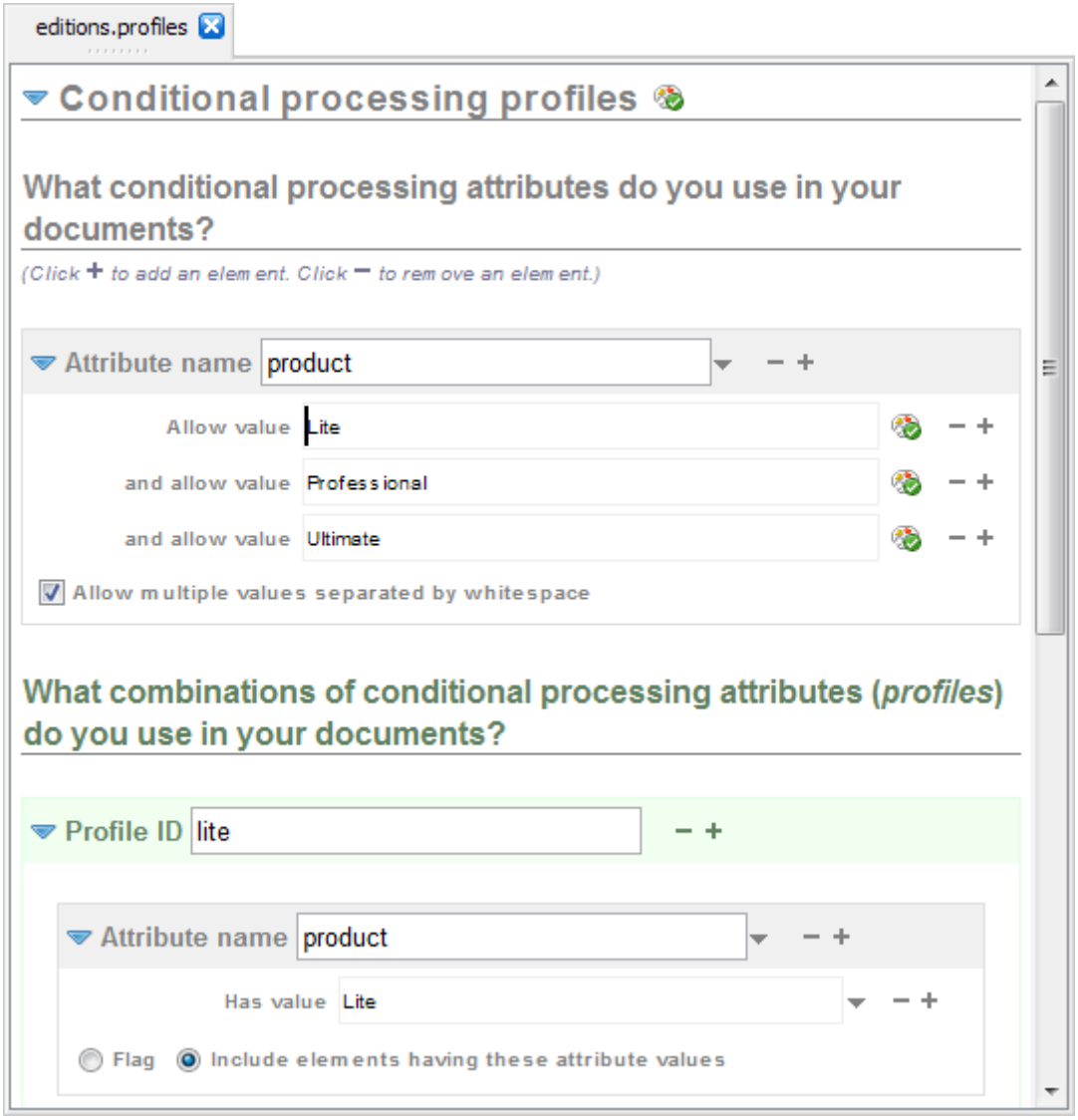
- The "**Style of Selected Profile**" is the style given to elements which are going to be *included* in the deliverables. An element is given this style if it has one or more of the profiling attributes referenced in the selected profile, each profiling attribute containing at least one of the attribute values referenced in the selected profile.
- The "**Style of Unselected Profile**" is the style given to elements which are going to be *excluded* from the deliverables. An element is given this style if it has one or more of the profiling attributes referenced in the selected profile, one or more of these profiling attributes *not* containing any of the attribute values referenced in the selected profile.

In general, using "**Set Style of Unselected Profile**" is all you'll have to do to get a clear idea of what will happen to the deliverables.

Both menu items display the the "**Set Style**" dialog box [11].

Select menu item "**Set Style of Unselected Profiles**" and use the "**Set Style**" dialog box to add a pink hatch background pattern to all elements matching any unselected profile.

Figure 1. The "`editions.profiles`" profile set containing the "`lite`", "`pro`" and "`ultimate`" profiles



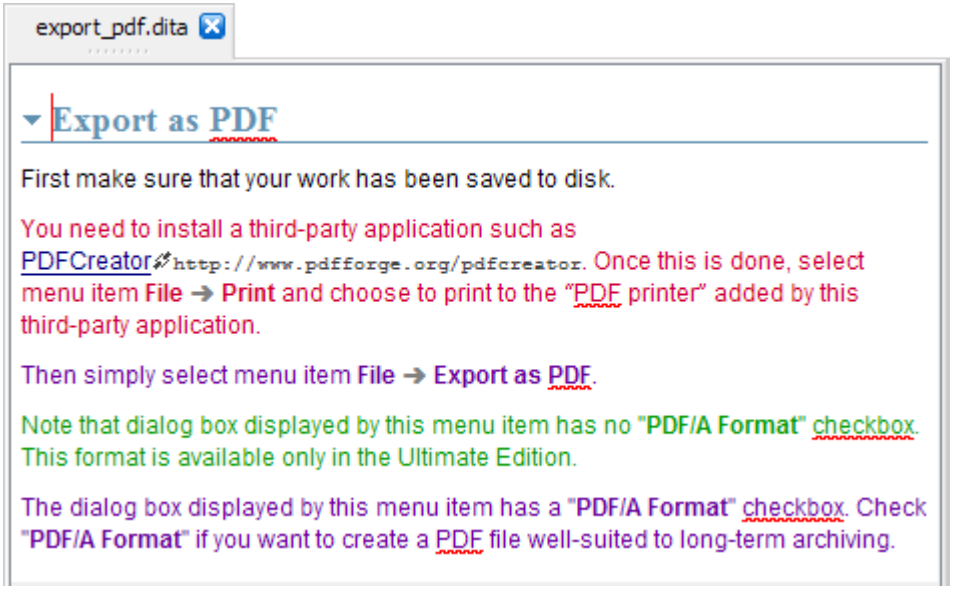
2. Save newly created profile set `editions.profiles` to disk.

3. Open `export_pdf.dita` and associate profile set `editions.profiles` to this DITA topic.

   This association is done using menu item **Topic** → **Conditional Processing** → **Select Profile**. This menu item displays the "**Select Profile**" dialog box [13]. This dialog box lets the user associate the document being edited to a set of profile with no specific profile selected or to a specific profile contained in a profile set.

   Given the styles specified in `editions.profiles`, the styled view of document `export_pdf.dita` will look like follows.

Figure 2. Styled view of `export_pdf.dita` after it has been associated to profile set `editions.profiles`



4. Set the `product` profiling attribute on paragraphs #2 to #4 of `editions.profiles` as shown in this screen [2].

   This is done using menu item **Topic** → **Conditional Processing** → **Set Profiling Attributes**. This menu item displays the "**Set Profiling Attributes**" dialog box [13] allowing to add, remove or replace one or more profiling attributes to/from the implicitly or explicitly selected element.

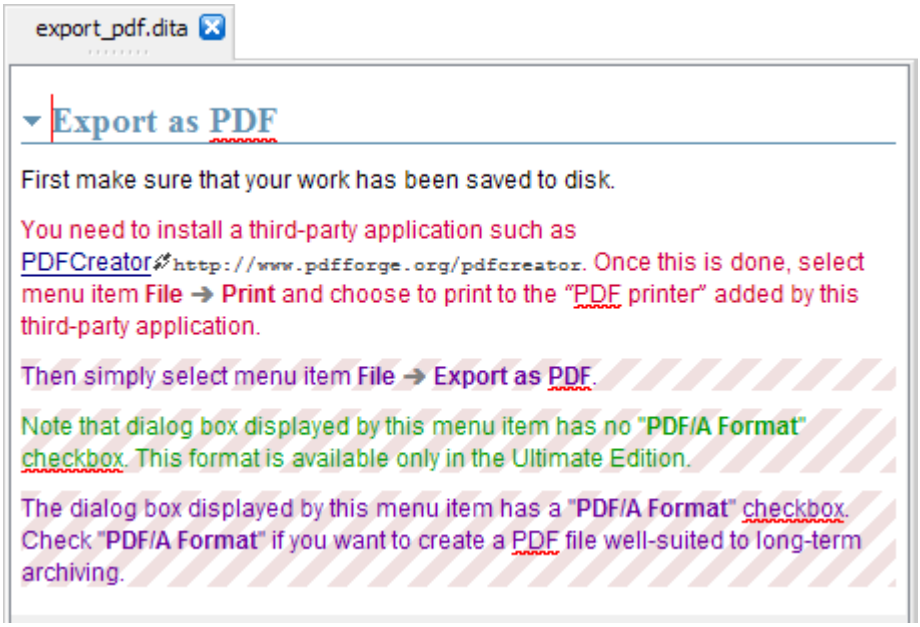
   Note that if you prefer to set the `product` attribute "by hand" using the **Attributes** tool and make a typo for an attribute value, this typo will be reported as a semantic warning by XMLmind XML Editor and an error message[2] will be displayed by the **Validity** tool.

5. Select the profile called "`lite`" using **Topic** → **Conditional Processing** → **Select Profile**.

   Given the styles specified in `editions.profiles`, the styled view of document `export_pdf.dita` will look like follows.

[2] Error message example: `"light", invalid value for conditional processing attribute "product".`

Figure 3. Styled view of `export_pdf.dita` when profile "`lite`" has been selected



6. Generate the PDF file corresponding to the Lite edition, normally, using **Topic** → **Convert Document** → **Convert to PDF**.

   There is no special parameter to pass to the DITA processor or to the XSLT stylesheet. If you want to generate the PDF file corresponding to the Professional edition, simply select the profile called "`pro`" using **Topic** → **Conditional Processing** → **Select Profile** and rerun **Convert to PDF**. Same procedure for generating the PDF file corresponding to the Ultimate edition.

3.1. Associating the "`.profiles`" file to the map rather than to individual topics

The sample map found in `sample/user_guide.ditamap` references just 2 topics `sample/export_pdf.dita` and `sample/advanced_features.dita`. What if this map contained hundreds of topics? Would the author have to associate the "`.profiles`" file to each of these topics? Fortunately the answer is no. Suffice to associate the "`.profiles`" file[3] to the map and to check **Tools** → **Use as Master Document** in *XMLmind XML Editor - Online Help*. When this is done, all the topics which are part of the deliverable created by this map automatically “inherit” the chosen conditional processing profile.

The same approach applies to DocBook 5.1 assemblies and to DocBook documents. For example, do not associate the "`.profiles`" file to each chapter document. Instead associate the "`.profiles`" file to the modular book.

The same approach applies to Ebooks. For example, do not associate the "`.profiles`" file to each ebook page. Instead associate the "`.profiles`" file to the ebook specification.

[3]This also works with "`.ditaval`" files.

4. About "`.profiles`" files

4.1. How it works

A "`.profiles`" file contain a specification of one or more conditional processing profiles in a proprietary format.

However it's important to always use XMLmind XML Editor to create and modify such files.

It's equally important to always use the right document template. For example, you cannot associate a "`.profiles`" file created using the "**DocBook v5+|Conditional Processing Profiles**" document template to a DITA map or topic. Only a "`.profiles`" file created using the "**DITA|Conditional Processing Profiles**" document template can be associated to a DITA map or topic.

The reason for these restrictions is that, each time you save a "`.profiles`" file to disk, XMLmind XML Editor automatically creates a *`profile_set_name`*`_files/` directory containing the following files:

- One *native implementation* per profile contained in the profile set. For example, when you save `editions.profiles` to disk, XMLmind XML Editor automatically creates `editions_files/lite.ditaval`, `editions_files/pro.ditaval` and `editions_files/ultimate.ditaval`.
- **DITA implementation only**: as of version 9, XMLmind XML Editor also automatically creates a simple subject scheme map file for use by **Map** → **Check Map**. For example, when you save `editions.profiles` to disk, XMLmind XML Editor also automatically creates `editions_files/attributeValues.ditamap`.

If you last saved your "`.profiles`" file using a version of **XXE** older than 9 and want to take advantage of this **Map** → **Check Map** enhancement, you'll have to re-save your existing "`.profiles`" file to disk so that the corresponding "`.ditaval`" filter files and "`.ditamap`" subject scheme map file are re-created behind the scene.

4.2. Creating a "`.profiles`" file allowing to flag contents in the deliverables

The example above shows how to mark contents displayed on screen by XMLmind XML Editor and how to exclude marked contents from the deliverables. However in some cases, you just want to *flag* (i.e. highlight) some contents in the deliverables, not exclude it.

This is possible too using a "`.profiles`" file, but only when this file has been created for/is associated to a DITA document.

Let's suppose you want to respectively strike out, give a green color and underline, elements having a `status` attribute respectively equals to `deleted`, `changed` and `new`.

1. Create a "`.profiles`" file using **File** → **New**, "**DITA|Conditional Processing Profiles**". Let's call this file `status.profiles`.

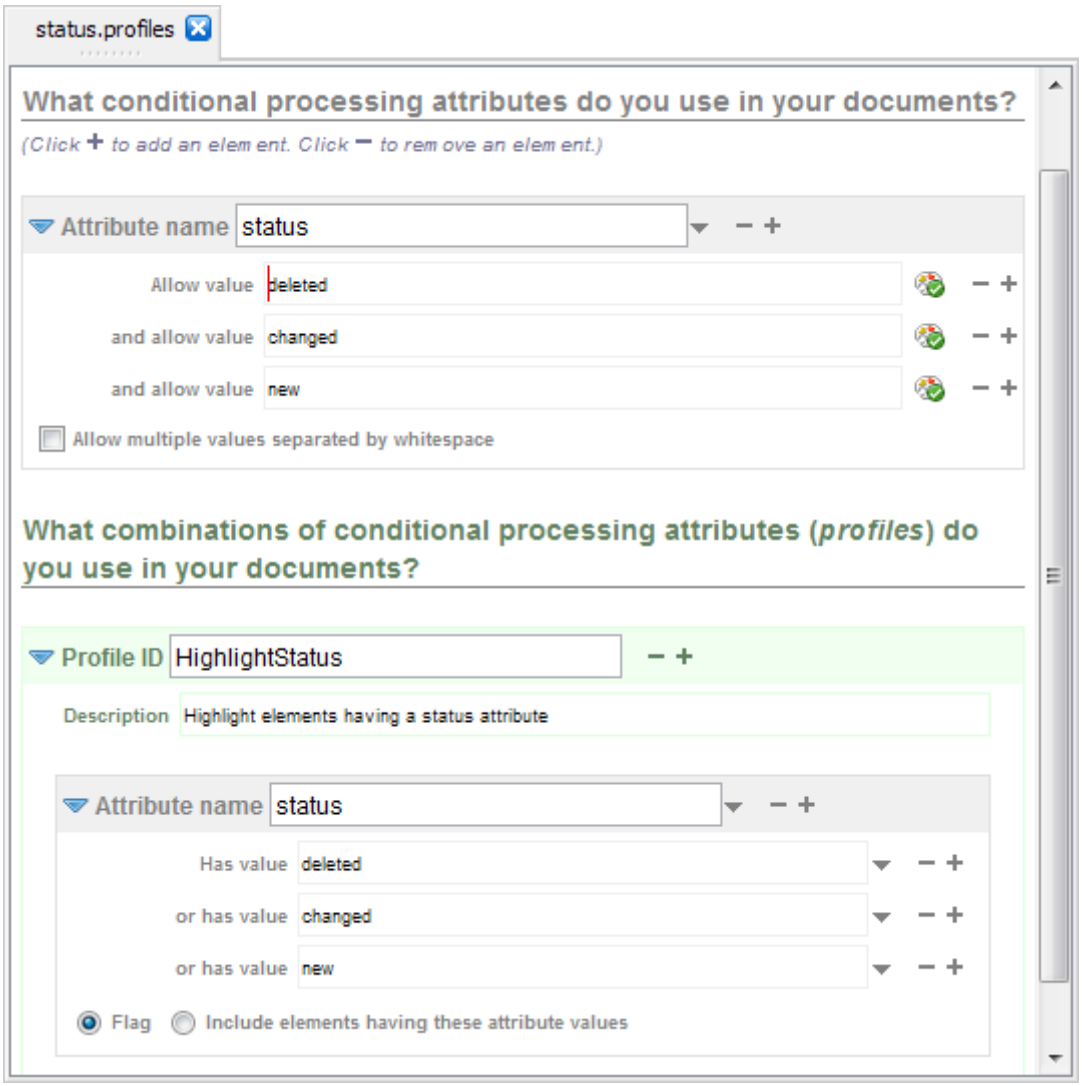
2. Declare `status` as being a profiling attribute and allow values `deleted`, `changed` and `new`.

3. Click the button which is next to each "**Allow value**" field. This button displays the "**Set Style**" dialog box [11]. Use this dialog box to give distinctive styles to `deleted`, `changed` and `new`; for example, strike out, green color and underline.

4. Define a profile making use of attribute `status` and *all its values*. For example, let's call this profile `HighlightStatus`.

5. Check **Flag**, the checkbox found below the list of "**Has value**" items. (Note that by default, it is **Include** and not **Flag** which is checked.)

   Figure 4. A sample "`.profiles`" file allowing to flag contents in the deliverables



6. Associate `status.profiles` and its profile `HighlightStatus` to your DITA map.

7. Doing this will not only mark elements having a `status` attribute in the styled views of the DITA map and all the referenced topics, but also flag these elements in the deliverables when you'll use **Convert Document** → **Convert to PDF**[4].

[4]This works because, as explained in Section 4.1, “How it works” [9], saving `status.profiles` automatically creates `status_file/HighlightStatus.ditaval`:

What's above is just a very simple example. It's of course possible to mix profiling attributes excluding contents and attributes flagging contents in the same profile.

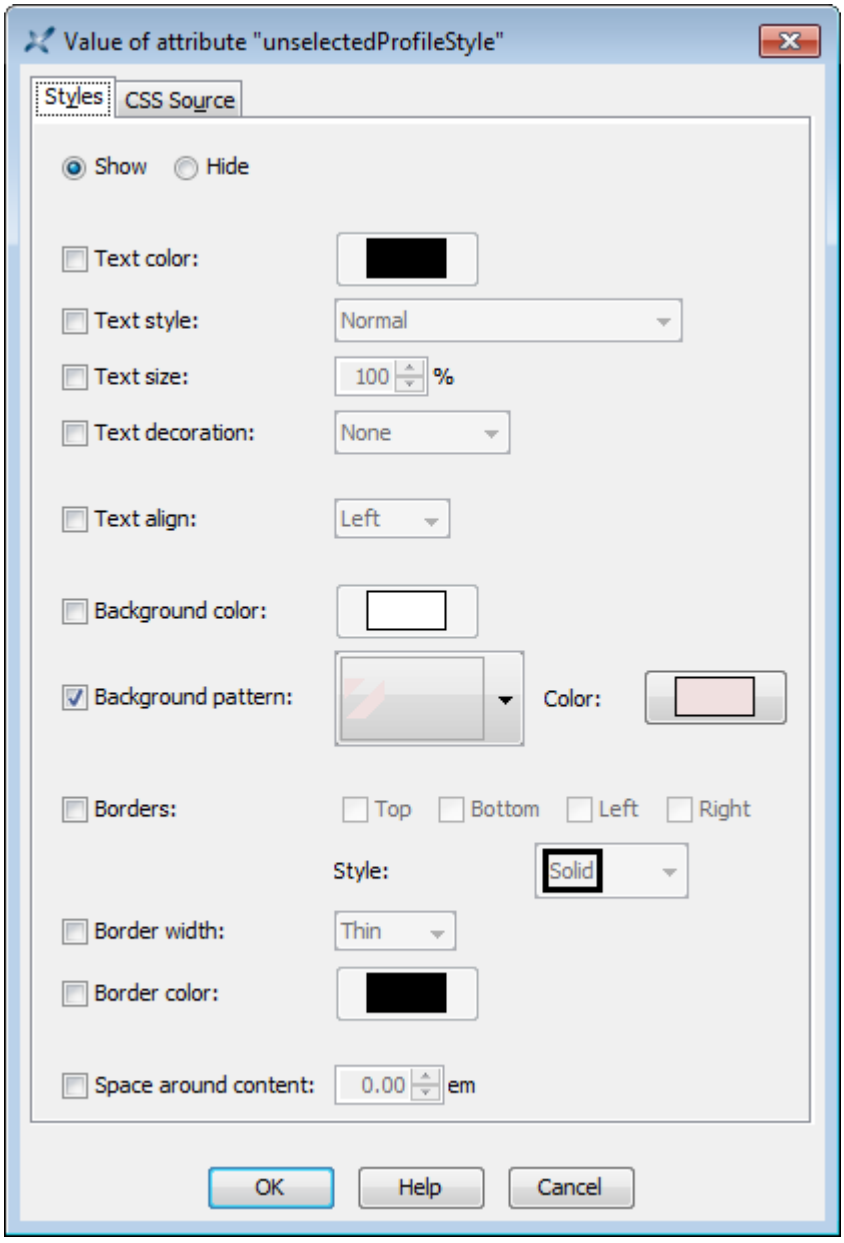
5. Dialog box reference

5.1. The "Set Style" dialog box

This dialog box is displayed when you click any of the "**Set style**" buttons found in the styled view of a "**Conditional Processing Profiles**" document. It can be used to specify which styles should be automatically added to elements having profiling attributes. This makes its easy spotting on screen which elements have been marked for conditional processing.

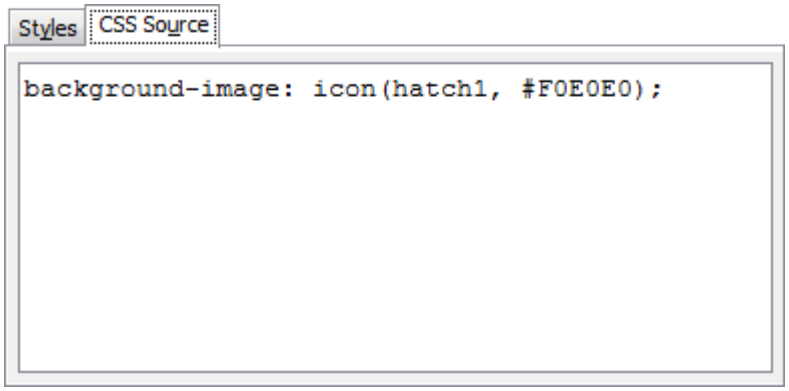
```
<val>
  <prop action="flag" att="status" color="#339900" val="changed"/>
  <prop action="flag" att="status" style="line-through" val="deleted"/>
  <prop action="flag" att="status" style="underline" val="new"/>
</val>
```

*Figure 5. The "**Set Style**" dialog box*



The dialog box depicted in the above screenshot is used to add a pink hatch background pattern to all “block elements” having all the profiling attribute values corresponding to any profile other than the selected one. This is a common way to spot which marked elements will not appear in the deliverable. Another common way to detect such excluded elements is to simply check "**Hide**".

*Figure 6. The "**CSS Source**" tab of the "**Set Style**" dialog box*

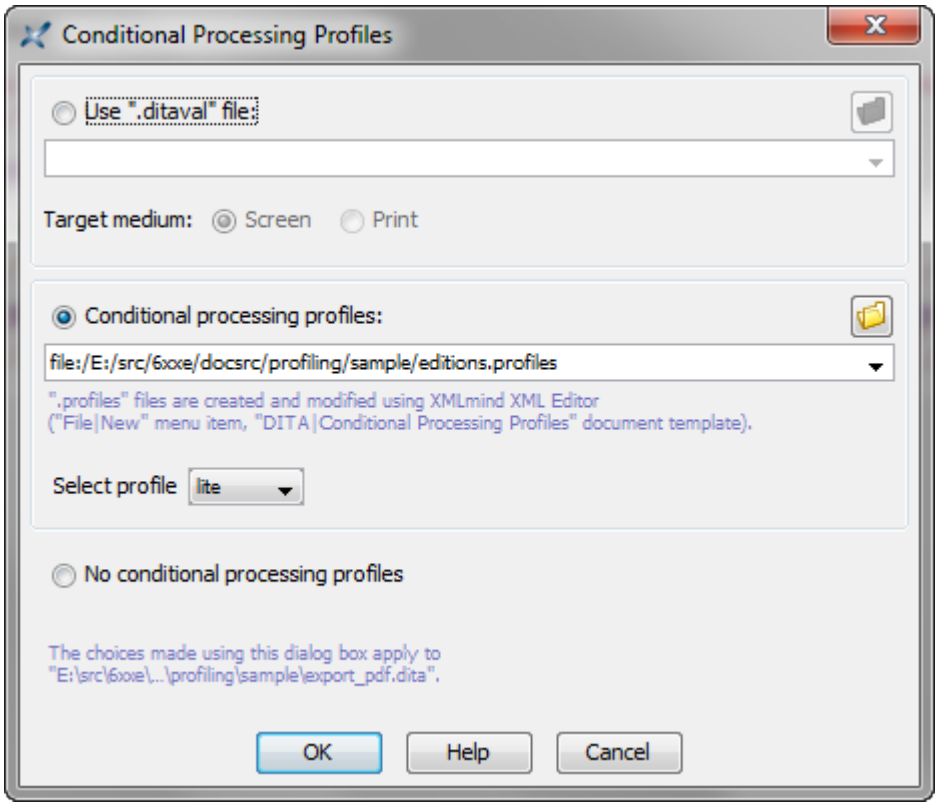


Selecting the "**CSS Source**" tab automatically updates the corresponding text area to reflect the choices made in the "**Styles**" tab. Selecting the "**Styles**" tab automatically updates the corresponding form to reflect the **CSS** properties typed in the "**CSS Source**" tab. Of course, with the "**CSS Source**" tab, you are not limited to using the few styles supported by the "**Styles**" tab.

5.2. The "Select Profile" dialog box

This dialog box is displayed by menu item ***XML menu*** → **Conditional Processing** → **Select Profile**.

*Figure 7. The "**Select Profile**" dialog box*



Note that the upper frame of the above dialog box, "**Use '.ditaval' file**:", is specific to DITA. This frame is absent if the document being edited is a DocBook document or an ebook specification.

This dialog box is used as follows:

- In order to associate a profile set (that is, a "`.profiles`" file) to the document being edited:

  1. Check "**Conditional processing profiles**".

  2. Choose a "`.profiles`" file previously created in XMLmind XML Editor.

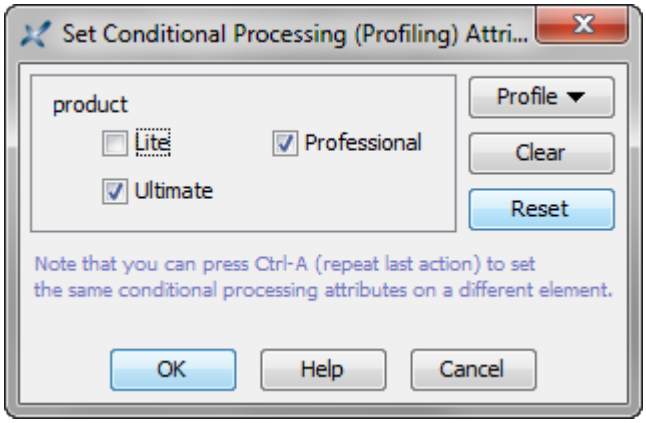
  3. Optionally, using the "**Select profile**" combobox, select one of the profiles contained in the chosen profile set.

- Check "**No conditional processing profiles**" to remove the association existing between a profile set and the document being edited.

5.3. The "Set Profiling Attributes" dialog box

This dialog box is displayed by menu item ***`XML menu`*** → **Conditional Processing** → **Set Profiling Attributes**.

*Figure 8. The "**Set Profiling Attributes**" dialog box*



This dialog box is available only when a profile set (that is, a "`.profiles`" file) has been associated to the document being edited. It displays all the profiling attributes referenced in the profile set. For each attribute, it displays all the values allowed in the profile set.

This dialog box is used as follows:

- Checking the checkbox corresponding to an attribute value adds this attribute value to the implicitly or explicitly selected element.
- Unchecking the checkbox corresponding to an attribute value removes this attribute value from the selected element.
- Other buttons:

  Profile

  Displays a menu containing the IDs of all the profiles contained in the profile set. Selecting a profile from such menu, automatically checks the checkboxes corresponding to this profile. For example, selecting profile "`lite`" unchecks all checkboxes except "**Lite**".

  Clear

  Unchecks all checkboxes.

  Reset

  Revert the dialog box to its initial state, that is, all the changes you made in the dialog box are undone.

A. Integrating the "Easy Profiling" add-on with document types other than DITA, DocBook and Ebook

Normal users are not supposed to do this. The intended audience for this appendix is consultants and local gurus.

What follows is a description of what has been done to integrate the "**Easy Profiling**" add-on with DocBook (v4). Integrating the "**Easy Profiling**" add-on with other document types should be very similar.

1. File *profiling_install_dir*/docbook/template.profiles contains a document template for a ".profiles" file which is specific to DocBook.

```
<profiles xmlns="http://www.xmlmind.com/xmleditor/schema/profiling"
          xmlns:prf="http://www.xmlmind.com/xmleditor/schema/profiling"
          class="com.xmlmind.xmleditext.profiling.docbook.DBProfileSet">

  <attributes>
    <attribute name="???">
      <value></value>
    </attribute>
  </attributes>

  <profile xml:id="???">
    <description xml:space="preserve"></description>

    <attribute name="???">
      <value></value>
    </attribute>
  </profile>

</profiles>
```

The only part of this document template which is specific to DocBook is attribute `class="com.xmlmind.xmleditext.profiling.docbook.DBProfileSet"`.

This attribute specifies that in the case of a DocBook document, abstract class `com.xmlmind.xmleditext.profiling.ProfileSet` is implemented by class `com.xmlmind.xmleditext.profiling.docbook.DBProfileSet`.

Class `DBProfileSet` has been written in the Java™ programming language. It consists in just a few lines of declarative code because it extends abstract class `com.xmlmind.xmleditext.profiling.XSLTProfileSet`.

```
public final class DBProfileSet extends XSLTProfileSet {
    ...

    @Override
    public char getDefaultSeparator() {
        return ';';
    }

    protected String[] getProfilingAttributeNames() {
        return PROFILING_ATTRIBUTE_NAMES;
    }

    protected String getParameterFormat() {
        return "profile.%s";
    }

    protected String getNameParameterFormat() {
        // A single custom profiling attribute is supported by the
        // DocBook XSLT stylesheets. Hence no "%d" in the format.
```

```
        return "profile.attribute";
    }

    protected String getValueParameterFormat() {
        return "profile.value";
    }
}
```

2. File *profiling_install_dir*/docbook/profiling.incl contains these additional configuration elements:

```
<configuration xmlns="http://www.xmlmind.com/xmleditor/schema/configuration"
  xmlns:cfg="http://www.xmlmind.com/xmleditor/schema/configuration">

  <include location="profiling:common/common.incl" />❶

  <template location="template.profiles"
            name="Conditional Processing Profiles"
            category="DocBook" order="2000" />❷

  <profiling>❸
    <class>com.xmlmind.xmleditext.profiling.docbook.DBProfileSet</class>
  </profiling>

  <menu label="-" insert="after _Demote">❹
    <separator />
    <menu label="Conditional Processin_g">
      <item label="_Select Profile..."
            command="setProfiling" />❺
      <item label="S_et Profiling Attributes..."
            command="setProfilingAttributes" />❻
    </menu>
  </menu>

</configuration>
```

❶ File common.incl declares the online help of the "**Easy Profiling**" add-on by the means of the help configuration element. It also declares how the "**Easy Profiling**" add-on is localized by the means of the translation configuration element.

❷ This template configuration element adds document template template.profiles (see above) to the dialog box displayed by **File** → **New**.

❸ This profiling configuration element specifies that the conditional processing (also called profiling) is to be implemented by class com.xmlmind.xmleditext.profiling.docbook.DBProfileSet in the case of DocBook documents. See Section 1, “The profiling configuration element” [18].

❹ This menu configuration element inserts in the **DocBook** menu, after the **Demote** menu item, submenu "**Conditional Processing**".

❺ First entry in the "**Conditional Processing**" is "**Select Profile**" and invokes command `setProfiling`. See Section 2.1, “Command `setProfiling`” [19].

❻ Second entry in the "**Conditional Processing**" is "**Set Profiling Attributes**" and invokes command `setProfilingAttributes`. See Section 2.2, “Command `setProfilingAttributes`” [19].

3. The above file is referenced in the stock DocBook configuration (*`XXE_install_dir`*`/addon/config/docbook/docbook.xxe`) as follows:

```
<!-- Do not report an error if the "Easy Profiling" add-on
     has not been installed. -->
<include location="---❶profiling:docbook/profiling.incl❷" />
```

❶ The "`---`" prefix before an URL instructs **XXE** to silently skip the inclusion when the URL cannot be successfully resolved.

❷ URL `profiling:docbook/profling.incl` resolves to *`profiling_install_dir`*`/docbook/profiling.incl` because XML catalog *`profiling_install_dir`*`/catalog.xml` contains the following rule:

```
<rewriteURI uriStartString="profiling:" rewritePrefix="." />
```

4. File *`profiling_install_dir`*`/common/profiling.imp` contains the following declaration:

```
@extension "com.xmlmind.xmleditext.profiling.StyleSheetExtension";
```

The above **CSS** stylesheet extension allows the automatic addition of styles to elements having profiling attributes. It can also automatically add styles to elements matching the selected profile and/or to elements matching any “unselected” profile.

5. The above file is referenced in the stock DocBook **CSS** stylesheet (*`XXE_install_dir`*`/addon/config/docbook/css/docbook.css`) as follows:

```
/* Do not report an error if the "Easy Profiling" add-on
   has not been installed. */
@import url(---profiling:common/profiling.imp);
```

1. The `profiling` configuration element

```
<profiling
  location = anyURI
>
  Content: [ class ]?
</profiling>

<class>
  Content: Java class name
</class>
```

This element specifies which class deriving from `com.xmlmind.xmleditext.profiling.ProfileSet` is to be used to implement conditional processing (also called *profiling*) for a given document type.

Optional attribute `location` specifies which set of profiles to automatically use by default[1] for all documents of a given document type. This location must point to a "`.profiles`" file created using XMLmind XML Editor.

If this location ends with a fragment, then this fragment must be the ID of a profile contained in the profile set. When such profile ID is specified, the corresponding profile is automatically selected by default[1] for all documents of a given document type.

DocBook example:

```
<profiling>
  <class>com.xmlmind.xmleditext.profiling.docbook.DBProfileSet</class>
</profiling>
```

DITA map example 1: by default, automatically use profile set "`file:/C:/docsrc/manual/products.profiles`" for all DITA maps.

```
<profiling location="file:/C:/docsrc/manual/products.profiles">
  <class>com.xmlmind.xmleditext.profiling.dita.DITAProfileSet</class>
</profiling>
```

DITA map example 2: by default, automatically use profile "`Lite`" contained in profile set "`file:/C:/docsrc/manual/products.profiles`" for all DITA maps.

```
<profiling location="file:/C:/docsrc/manual/products.profiles#Lite">
  <class>com.xmlmind.xmleditext.profiling.dita.DITAProfileSet</class>
</profiling>
```

2. Profiling commands

2.1. Command `setProfiling`

No parameter.

This command displays a dialog box [13] allowing to associate a set of profiles or a profile belonging to a profile set to the document being edited. It also lets the user remove the association existing between the document being edited and a profile or a set of profiles.

2.2. Command `setProfilingAttributes`

Parameter syntax:

```
[ profiling_attribute_name S profiling_attribute_value
  [ S profiling_attribute_name S profiling_attribute_value ]* ]
```

This command is disabled unless the document being edited has been associated to a set of profiles.

[1] This can be overridden for specific documents using "***XML** menu* → **Conditional Processing** → **Select Profile**".

Without a parameter, this command displays a dialog box [13] allowing to add, replace or remove one or more profiling attributes to/from the implicitly or explicitly selected element.

When passed an empty string as a parameter, this command removes all the profiling attributes from the selected element.

When passed a parameter string containing (possibly quoted) attribute name/value pairs, the command first removes all the profiling attributes from the selected element if any, and then, adds specified profiling attributes to the selected element. This facility is mainly useful to repeat the command (**Edit** → **Repeat**; keyboard shortcut **Ctrl**+**A**) on another element.

Examples:

```
setProfilingAttributes
setProfilingAttributes ""
setProfilingAttributes userlevel advanced security "for your eyes only"
```