
XOOPS Documentation With DocBook

by Xoops Documentation Team

XOOPS Documentation With DocBook

by Xoops Documentation Team

This document works as a miniguide to docBook, and provides a sample template document for XOOPS.

released under the Creative Commons license

Table of Contents

1. Miniguide for DocBook	1
1.1. Introduction	1
1.2. What files do we have?	2
1.3. The master document file	2
2. Xoops samples	5
2.1. The first section	5
2.2. The first document sample	5
2.3. A sample for a image callout	5
2.4. A callout sample	6
2.5. An FAQ example	7
2.6. DocBook resources	8
3. My first chapter	9
3.1. My first section within this chapter	10
3.2. My external section	11
4. Installing DocBook XML	13
4.1. Java	13
4.2. Install the docbook DTD	13
4.3. Install the docbook XSL	13
4.4. Install libxml	13
4.5. Install Saxon	13
4.6. Install FOP	13
4.7. Install Jai	14
4.8. Install Jimi	14
4.9. Install catalog resolver for Saxon	14
4.10. Install HtmlHelp	14
4.11. Install Ant	14
4.12. Install Ant contribution packet (ant-contrib-0.6, did not yet try a newer release)	14
4.13. PHP cli	15
4.14. How to edit XML	15
5. The Build Process	17
5.1. A sample build batch file	17
5.2. The ant file	17
6. Directory Structure	19
7. jEdit for XML-editing	21
7.1. Installing Plugins	21
7.2. Configure jEdit	21
7.2.1. Choose a font	21
7.2.2. Choose the colors	21
7.2.3. Editing Options	21
7.2.4. A Sample View of jEdit	22
Index	25



List of Figures

2.1. User Info Settings	5
2.2. A handmade callout picture	6
3.1. This is just a sample figure	10
7.1. Open the Plugins manager	21
7.2. jEdit settings for editing options	22
7.3. A Sample View of jEdit	23



List of Tables

3.1. A table can have a title	12
-------------------------------------	----



Chapter 1. Miniguide for DocBook

Abstract

This miniguide should be the first help for people with no experience in DocBook to write simple DocBook documents.

1.1. Introduction

This miniguide provides some small sample DocBook XML files which uses the most used tags within DocBook documents.

Writing DocBook XML files could be done simply with normal text editor programs. Writing an XML file is nearly the same as writing a plain¹ HTML file.



Note

So everyone who has ever written a plain HTML file, should be familiar with XML at once. The only difference of course is, that with DocBook XML we have different tags.

In HTML you write the following:

```
<p>This is a simple paragraph</p>
```

In DocBook it looks like the following:

```
<para>This is a simple paragraph</para>
```

Or an example for a list:

```
<ul>
  <li>something here</li>
  <li>and here</li>
</ul>
```

in DocBook (ok, it looks a little more complicated than in HTML):

```
<itemizedlist>
  <listitem><para>something here</para></listitem>
  <listitem><para>and here</para></listitem>
</itemizedlist>
```

The DocBook DTD (Document Type Definition) declares only the allowed tags, and where they can occur. The same is true for XHTML by the way (it's only a different DTD).

The idea of DocBook, and of XML in general, is to assign your content with more specific information, which leads to an abstraction between data (content) and layout (how it is shown)

You don't mark your documents in this way (shown in a XML syntax):

```
This is my program <b>c:program Files/myProgram.exe</b>
  and this is my database <em>MyDatabase</em>
  and this is the author: <b>Billy the Gates</b>.
```

but in this way:

```
This is my program <filename>c:program Files/myProgram.exe</filename>
```

¹Plain means without any help of a WYSIWYG HTML-editor

```
and this is my database <database>MyDatabase</database> and this is the au
<author><firstname>Billy</firstname><surname>the Gates</surname></author>
```

This approach allows you to concentrate on the content, and to control the layout with a global customization layer. So you decide later how the information should be shown to the reader.

1.2. What files do we have?

In general our document are split into some smaller files. We have a master document file and some chapter or section files which will be included by the master document file². This include files are located within a subdirectory. The complete document will be merged temporary only at conversion time (when the different output formats are generated).

This document for example consists of the following files:

```
docroot/src/en/XD-999.xml
docroot/src/en/docbook/mg_intro.xml
docroot/src/en/docbook/mg_sample_chapter.xml
docroot/src/en/docbook/mg_sample_section.xml
```

1.3. The master document file

The master document file, in our example `XD-999.xml`, is a short file which has, besides some general document information, some include statements. The following shows a simple form of this file. Everthing that's between the `<bookinfo> ... </bookinfo>` whithin the original file shouldn't bother us at the moment. The same is true for the `<index ...> </index>`

```
<?xml version="1.0" encoding='utf-8'?>
<!-- jEdit buffer-local properties: -->
<!-- :indentSize=2:noTabs=true: -->
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
"http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd" [
  <!ENTITY mg_sample_section SYSTEM "mg_sample_section.xml">
  <!ENTITY sample SYSTEM "sample.xml">
  <!ENTITY mg_intro SYSTEM "mg_intro.xml">
  <!ENTITY mg_sample_chapter SYSTEM "mg_sample_chapter.xml">
  <!ENTITY installation SYSTEM "installation.xml">
  <!ENTITY % glossary SYSTEM "../glossary.ent">
  %glossary;
]>
<book lang="en">

  &mg_intro;
  &sample;
  &mg_sample_chapter;
  &installation;

  <index role="with_index" id="TheVeryLastPage">
  </index>
</book>
```

For the startup DocBook writer the only interest here are the include statements. We include files as file entities

The file entities are included in the normal entity way.

```
&sample;
```

What is now included by the first file? It's just exactly this chapter, the lines you just read. To make things a little easier, we will continue with the second include statement, which includes the `docbook/sample.xml`. With-

²This is nothing special to DocBook, but indeed a good idea in general.

in this chapter we will explain the most common DocBook tags.



Chapter 2. Xoops samples

2.1. The first section

To show the possibilities of docbook, I have written this sample document. I hope it helps you to get a little familiar with “DocBook”

In chapter Chapter 4, *Installing DocBook XML* [13] there is all the installation related information about how to install the necessary DocBook things.

At the end of this chapter, there is a section (Section 2.6, “DocBook resources” [8]) with some resources about DocBook.

Of course, it could be that the Xoops documentation group will decide to not use DocBook, because it seems to be too complicated, too much new things, the group began already with something different and you don't want to change now. May be I can take a document you have already written, and do a conversion to DocBook?

2.2. The first document sample

This sample shows a picture of the “User Info Settings”.

Preferences Main >>> User Info Settings

User Info Settings	
Allow new user registration? Select yes to accept new user registration	<input checked="" type="radio"/> Yes <input type="radio"/> No
Minimum length of password required	<input type="text" value="5"/>
Minimum length of username required	<input type="text" value="5"/>
Maximum length of username	<input type="text" value="10"/>
Allow users to change email address?	<input type="radio"/> Yes <input checked="" type="radio"/> No
Notify by mail when a new user is registered?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Select group to which new user notification mail will be sent	<input type="text" value="Registered Users"/>
Select activation type of newly registered users	<input type="text" value="Requires activation by user (recommended)"/>
Select group to which activation mail will be sent <small>Valid only when 'Activation by administrators' is selected</small>	<input type="text" value="Webmasters"/>

Figure 2.1. User Info Settings

2.3. A sample for a image callout

Unfortunately image callouts don't work yet. This means, the marking pictures aren't merged into the picture which is to explained. This would really be a nice feature when explaining graphical user interfaces. To work around this, it would be necessary to create such a merged picture by hand. See Figure 2.2, “A handmade callout picture” [6] for a sample picture.

Preferences Main >>> User Info Settings

User Info Settings	
Allow new user registration? Select yes to accept new user registration	<input checked="" type="radio"/> Yes <input type="radio"/> No
Minimum length of password required	<input type="text" value="5"/>
Minimum length of username required	<input type="text" value="5"/>
Maximum length of username	<input type="text" value="10"/>
Allow users to change email address?	<input type="radio"/> Yes <input checked="" type="radio"/> No
Notify by mail when a new user is registered?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Select group to which new user notification mail will be sent	<input type="text" value="Registered Users"/>
Select activation type of newly registered users	<input type="text" value="Requires activation by user (recommended)"/>
Select group to which activation mail will be sent <small>Valid only when 'Activation by administrators' is selected</small>	<input type="text" value="Webmasters"/>

- ❶ Where does it point to?
- ❷ This also point to nowhere

Preferences Main >>> User Info Settings

User Info Settings	
Allow new user registration? Select yes to accept new user registration	<input checked="" type="radio"/> Yes <input type="radio"/> No ❶
Minimum length of password required	<input type="text" value="5"/> ❷
Minimum length of username required	<input type="text" value="5"/> ❸
Maximum length of username	<input type="text" value="10"/> ❹
Allow users to change email address?	<input type="radio"/> Yes <input checked="" type="radio"/> No
Notify by mail when a new user is registered?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Select group to which new user notification mail will be sent	<input type="text" value="Registered Users"/>
Select activation type of newly registered users	<input type="text" value="Requires activation by user (recommended)"/>
Select group to which activation mail will be sent <small>Valid only when 'Activation by administrators' is selected</small>	<input type="text" value="Webmasters"/>

Figure 2.2. A handmade callout picture

2.4. A callout sample

```
SET PATH=C:\docbook\tools;%PATH% ❶
SET XML_CATALOG_FILES=C:\docbook\docbook-xml-4.2\catalog
xmllint --xinclude XoopsSampleDoc.xml > XoopsSampleDocResolved.xml ❷

@ECHO OFF
```

```

set LOCALCLASSPATH=c:/docbook/docbook-xsl/extensions/saxon651.jar1
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:/docbook/saxon/saxon.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:/docbook/saxon/saxon_fop.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:/docbook/saxon/saxon_jdom.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:/docbook/docbook-xsl/extensions/resolver.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:/docbook/docbook-xsl/extensions

set SAXON=java -cp %LOCALCLASSPATH% com.icl.saxon.StyleSheet2
set SAXON=%SAXON% -x org.apache.xml.resolver.tools.ResolvingXMLReader
set SAXON=%SAXON% -y org.apache.xml.resolver.tools.ResolvingXMLReader
set SAXON=%SAXON% -u
set SAXON=%SAXON% -o ../PDF/XoopsSampleDocResolved.fo
set SAXON=%SAXON% XoopsSampleDocResolved.xml xoopscustom.xsl

@ECHO ON

%SAXON% >saxon.log3

@ECHO OFF

set LOCAL_FOP_HOME=c:/docbook/fop/4

set LIBDIR=%LOCAL_FOP_HOME%lib
set LOCALCLASSPATH=%LOCAL_FOP_HOME%build/fop.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;%LIBDIR%/xml-apis.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;%LIBDIR%/xercesImpl-2.2.1.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;%LIBDIR%/xalan-2.4.1.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;%LIBDIR%/batik.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;%LIBDIR%/avalon-framework-cvs-20020806.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:/docbook/jimi/jimiProClasses.zip
REM set LOCALCLASSPATH=%LOCALCLASSPATH%;%LIBDIR%/jai_core.jar
REM set LOCALCLASSPATH=%LOCALCLASSPATH%;%LIBDIR%/jai_codec.jar

set FOP=java -cp %LOCALCLASSPATH% org.apache.fop.apps.Fop5
set FOP=%FOP% -fo ../PDF/XoopsSampleDocResolved.fo
set FOP=%FOP% -pdf ../PDF/XoopsSampleDoc.pdf
set FOP=%FOP% -c c:/docbook/fop/conf/userconfig.xml

@ECHO ON

%FOP% >fop.lop6

```

- ¹ This is normally a part of the autoexec.bat, but due to the restricted size of the path variable in Win98, I put it here.
- ² This command merges all the single XML files to one big XML file. XML, per se, don't (yet) allow to divide files and include them into one master document.
- ³ Executes the conversion of the resolved XML file to the fo file.
- ⁴ This and the following lines form the LOCALCLASSPATH environment variable, which is necessary for FOP.
- ⁵ This and the following lines form the command line for calling FOP
- ⁶ Executes the conversion of the fo file to the PDF file

2.5. An FAQ example

This section shows an example of how a Question And Answer is done.

Is Xoops a good CMS?

No, it's a *very* good CMS.

Is this the second question?

No this is the second answer.

I have no idea of good question, what shall I do?

Sorry, I also have no idea for a good answer.

Is it necessary to use a CMS?

Is it necessary to eat?

What should I ask

What should I answer

What day is today

It depends on when you read this.

Is Max a psychopathic?

If it's Max from Tennessee, of course.

2.6. DocBook resources

The following list gives an overview about some important internet resources for DocBook.

- DocBook: The Definitive Guide [<http://www.docbook.org/tdg/en/html/docbook.html>]. This is the DocBook book from the creator of DocBook. It describes mainly the DocBook elements (tags).
- DocBook XSL: The Complete Guide [<http://www.sagehill.net/docbookxsl/index.html>]. This is a description of what could we do to get the output we like to have.
- <http://docbook.org/wiki/moin.cgi/WhyDocBook>. This wiki page should give you some arguments for why to use DocBook. Also the frontpage of this wiki site gives you a lot of other resources.

Chapter 3. My first chapter

The following lines show you the content of this (source)file including this paragraph.

```
<?xml version="1.0" encoding="utf-8"?> ❶
<!-- jEdit buffer-local properties: --> ❷
<!-- :indentSize=2:noTabs=true: -->
<!-- :xml.root=docmaster.xml: -->
<chapter id="cpt.mychapter_1"> ❸
  <title>My first chapter</title> ❹
  <para>The following lines shows you the content of this (source)file
  including this paragraph.</para> ❺
```

- ❶ The general XML header.
- ❷ Some jEdit properties set within a xml comment.
- ❸ The start tag of our chapter. As an attribute we give that chapter an id, which we can use for referencing this chapter.
- ❹ The title for this chapter.
- ❺ Our first paragraph within this chapter.

The next listing shows you an example of the admonition possibilities of DocBook. I think there is nothing to explain here. Just see at the listing, and how the output looks like.

```
<warning><para>Be careful, if it rains, you can get wet!</para></warning>
<note><para>Don't know whether the warning is really necessary</para></note>
<important><para>But this is very important</para></important>
<tip><para>Give me a tip</para></tip>
```



Warning

Be careful, if it rains, you can get wet!



Note

Don't know whether the warning is really necessary



Important

But this is very important



Tip

Give me a tip

The next listing shows an example of an figure, and a crossreference to the figure.

```
<para>The <xref linkend="fig.mysample_figure"/>
show us a HAPPY NEW YEAR picture.</para>

<figure id="fig.mysample_figure">
  <title>This is just a sample figure</title>
  <mediaobject>
    <imageobject>
      <imagedata align="center"
        fileref="img/en/docbook/new_year.png" format="PNG"/>
```

```

    </imageobject>
  </mediaobject>
</figure>

```

The Figure 3.1, “This is just a sample figure” [10] shows us a HAPPY NEW YEAR picture.



Figure 3.1. This is just a sample figure

Now let's go into a new section, and how we can achieve this.

```

<sect1 id="sec.myfirst">
  <title>My first section within this chapter</title>
  <para>A simple paragraph within a section</para>

```

3.1. My first section within this chapter

A simple paragraph within a section

With the following we can something place into an index.

```

<para>Something we put into the index. Lions are big cats
<indexterm><primary>Animal</primary><secondary>lion</secondary></indexterm>.<

```

Something we put into the index. Lions are big cats .

And now a list in which we incorporate some usefull tags.

```

<itemizedlist>
  <title>Some examples of the most important tags</title>
  <listitem>
    <para>To mark a filename <filename>MyFilename.txt</filename></para>
  </listitem>
  <listitem>
    <para>To mark a directory
    <filename class="directory">c:\this\is\the\path</filename></para>
  </listitem>
  <listitem>
    <para>To mark a database <database>MyDatabase</database></para>
  </listitem>
  <listitem>
    <para>A simple Link <ulink url="http://java.sun.com/products/jimi/#">
    </ulink></para>
  </listitem>
  <listitem>
    <para>A simple Link with a name
    <ulink url="http://java.sun.com/products/jimi/#">Jimi</ulink></para>
  </listitem>
  <listitem>
    <para>We make <emphasis>this</emphasis> italic</para>
  </listitem>
  <listitem>
    <para>If we want to show the user he/she should click on
    <guibutton>Next</guibutton></para>
  </listitem>
  <listitem>
    <para>You can see a simple table in <xref linkend="sec.mysecond"/></para>
  </listitem>

```

```
</itemizedlist>
```

Some examples of the most important tags

- To mark a filename `MyFilename.txt`
- To mark a directory `c:\this\is\the\path`
- To mark a database **MyDatabase**
- A simple Link `http://java.sun.com/products/jimi/#`
- A simple Link with a name Jimi [`http://java.sun.com/products/jimi/#`]
- We make *this* italic
- If we want to show the user he/she should click on **[Next]**
- You can see a simple table in Section 3.2, “My external section” [11]

And now the end of this section and before the end of this chapter, there is again a entity for including a file. The next section then explains the content of the `docbook/mg_sample_section.xml` and gives some additional examples.

```
</sect1>
<!-- This will include a external section file -->
&mg_sample_section;
</chapter>
```

3.2. My external section

Again, just a simple paragraph, and after the paragraph an ordered list.

The next examples show you an ordered list with numbers and an ordered list with letter

```
<para>
  <orderedlist>
    <!-- The title could also be omitted -->
    <listitem><para>This is the first item</para></listitem>
    <listitem><para>This is the second item</para></listitem>
    <listitem><para>This is the third item</para></listitem>
  </orderedlist>
</para>
<!-- The list hasn't to be within a para -->
<orderedlist numeration="loweralpha">
  <!-- The title could also be omitted -->
  <listitem><para>This is the first item</para></listitem>
  <listitem><para>This is the second item</para></listitem>
  <listitem><para>This is the third item</para></listitem>
</orderedlist>
```

1. This is the first item
2. This is the second item
3. This is the third item

- a. This is the first item
- b. This is the second item
- c. This is the third item

And again something for the index page.

```
<para>
  Again something for the index. Tiger are big cats<indexterm><primary>Animals
</para>
<para>Something more complicated... a simple table<indexterm><primary>table</para>
```

Again something for the index. Tiger are big cats.

Something more complicated... a simple table.

And now a simple table

```
<table>
  <title>A table can have a title</title>
  <tgroup cols="4">
    <colspec align="right" colwidth="*" />
    <colspec align="left" colwidth="2*" />
    <colspec align="center" colwidth="3*" />
    <colspec align="right" colwidth="*" />
    <thead>
      <row>
        <entry>number</entry>
        <entry>a simple text</entry>
        <entry>more text</entry>
        <entry>value</entry>
      </row>
    </thead>
    <tbody>
      <row>
        <entry>1</entry>
        <entry>second</entry>
        <entry>something to the center</entry>
        <entry>1234</entry>
      </row>
      <row>
        <entry>2</entry>
        <entry>third</entry>
        <entry>less</entry>
        <entry>99887766</entry>
      </row>
    </tbody>
  </tgroup>
</table>
```

number	a simple text	more text	value
1	second	something to the center	1234
2	third	less	99887766

Table 3.1. A table can have a title

Chapter 4. Installing DocBook XML

A good tutorial for installing Docbook is available at:

<http://supportweb.cs.bham.ac.uk/documentation/tutorials/docsystem/build/tutorials/tutorialshome.html>

4.1. Java

For conversion process with Saxon und FOP we need a Java runtime environment. This is available at:

<http://java.sun.com/j2se/downloads.html>

4.2. Install the docbook DTD

The docbook DTD could be downloaded from the following link:

<http://www.docbook.org/xml/4.2/docbook-xml-4.2.zip>

Unzip to something like: `c:\docbook\docbook-xml-4.2`

4.3. Install the docbook XSL

The docbook XLS could be downloaded from the following link:

http://sourceforge.net/project/showfiles.php?group_id=21935

Unzip it to something like: `c:\docbook\docbook-xsl`

We have to make one modification to `htmlhelp-common.xsl`; open the file and search for “[FILES]”; right after this line include a line with the following text “`img/nav/*.gif`”. This is necessary so that the `htmlhelp` compiler can search the images for navigation in the given directory.

4.4. Install libxml

The windows binaries for libxml are available at the following URL:

<http://www.zlatkovic.com/libxml.en.html>

It's the best to use the FTP link:

<ftp://ftp.zlatkovic.com/pub/libxml/>

There you can download the most recent versions of `iconv`, `libxml2` and `libxslt`.

Unzip the files, and move the `exe` and `dll` files into something like: `c:\docbook\tools`

4.5. Install Saxon

Saxon is used to convert the docbook XML file to a `fo` file. `fo` is a Formatting Objects XML file. A `fo` file is the basis for the conversion to PDF.

Saxon can be downloaded at:

<http://saxon.sourceforge.net/>

Unzip it to something like: `c:\docbook\saxon`

4.6. Install FOP

FOP is used to convert a fo file to PDF. There are also some other output formats available, but PDF is the best supported conversion format at the moment. FOP could be downloaded from the following link:

<http://www.apache.de/dist/xml/fop/>

Unzip it to something like: `c:\docbook\fop`

4.7. Install Jai

To use also PNG pictures within our PDF documents, we have to download jai. Download “Windows CLASSPATH Install” from the following site:

http://java.sun.com/products/java-media/jai/downloads/download-1_1_2.html

Run the exe and choose `c:\docbook\jai` as installation directory.

4.8. Install Jimi

A different library for using images is Jimi. This is available at:

<http://java.sun.com/products/jimi/#>

Unzip the `jimix_x.zip` within `c:\docbook`. The resulting path should look as `c:\docbook\jimi\JimiProClasses.zip`.

4.9. Install catalog resolver for Saxon

To resolve the DTD url to a local file when using Saxon, we need to download the `resolver.jar`: The `resolver.jar` can be downloaded at:

<http://xml.apache.org/commons/dist/>

We copy this file into the `c:\docbook\docbook-xsl\extensions`

4.10. Install HtmlHelp

The `HtmlHelp` compiler could be downloaded from:

<http://msdn.microsoft.com/library/en-us/htmlhelp/html/hwMicrosoftHTMLHelpDownloads.asp>

4.11. Install Ant

Ant is a build system from the apache group:

<http://http://ant.apache.org/>

Unzip it, and copy it to something like `c:\docbook\ant`. We should have now something like `c:\docbook\ant\bin` and `c:\docbook\ant\lib` etc..

4.12. Install Ant contribution packet (ant-contrib-0.6, did not yet try a newer release)

A packet for ant which adds some useful functions to ant:

http://http://sourceforge.net/project/showfiles.php?group_id=36177

Unzip it and copy `ant-contrib-0.6.jar` into `c:\docbook\ant\lib`

4.13. PHP cli

Install PHP and copy the PHP cli executable into the PHP main directory as *phpcli.exe*.

4.14. How to edit XML

There are several XML editors available. Some comercial and also some open source programs. I, for myself use jEdit which is available at:

<http://www.jedit.org>

For setting up and working with jEdit see Chapter 7, *jEdit for XML-editing* [21]



Chapter 5. The Build Process

5.1. A sample build batch file

To start the build process you can take the following listing and paste it into a `build.bat` and store it in `xoops/documents`. If you have installed the programs and files into different directories please make the proper adjustments.

```
SET PATH=C:\docbook\tools;%PATH%
SET PHP_HOME=C:/Programme/apache/friends/xampp/php
SET XML_CATALOG_FILES=C:\docbook\docbook-xml-4.2\catalog
SET JAVA_HOME=C:\programme\java\jdk1.5.0
SET HTMLHELP_HOME=C:\Programme\MVS\HTMLHelp

SET RECODE_HOME=C:/programme/gnu/iconv/iconv/util
SET INPUT_ENCODING=UTF-8

SET OUTPUT_ENCODING=ISO-8859-1

c:\docbook\ant\bin\ant -l ant.log -v
```

5.2. The “ant” file

The file which is responsible for controlling the build process is `build.xml`. Ant reads the file and processes according the dependencies which are chained up within this file.



Chapter 6. Directory Structure



Chapter 7. jEdit for XML-editing

7.1. Installing Plugins

Within the Installation chapter see Section 4.14, “How to edit XML” [15] from where you can download jEdit.

After you have downloaded and installed the packet start jEdit. Now we need to install the XML plugin. Without this plugin, we could have used nearly any other text editor, but this plugin makes jEdit as one of the best XML-editors available.

To install this (and every other plugin) go to the plugin manager as shown in Figure 7.1, “Open the Plugins manager” [21]. This will open the plugin manager dialog. Within this dialog click on the **[Install]** -tab and click into the checkbox for the XML-plugin. Other plugins which I would recommend are:

- “ErrorList”
- “SideKick”
- “XML Indenter”
- “WhiteSpace”
- “BufferTabs”

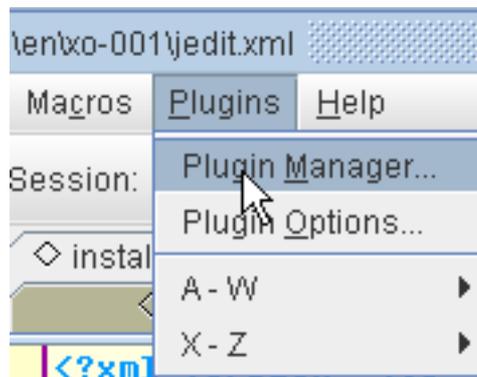


Figure 7.1. Open the Plugins manager

7.2. Configure jEdit

7.2.1. Choose a font

What font should we select?

7.2.2. Choose the colors

What are good colors

7.2.3. Editing Options

Setting

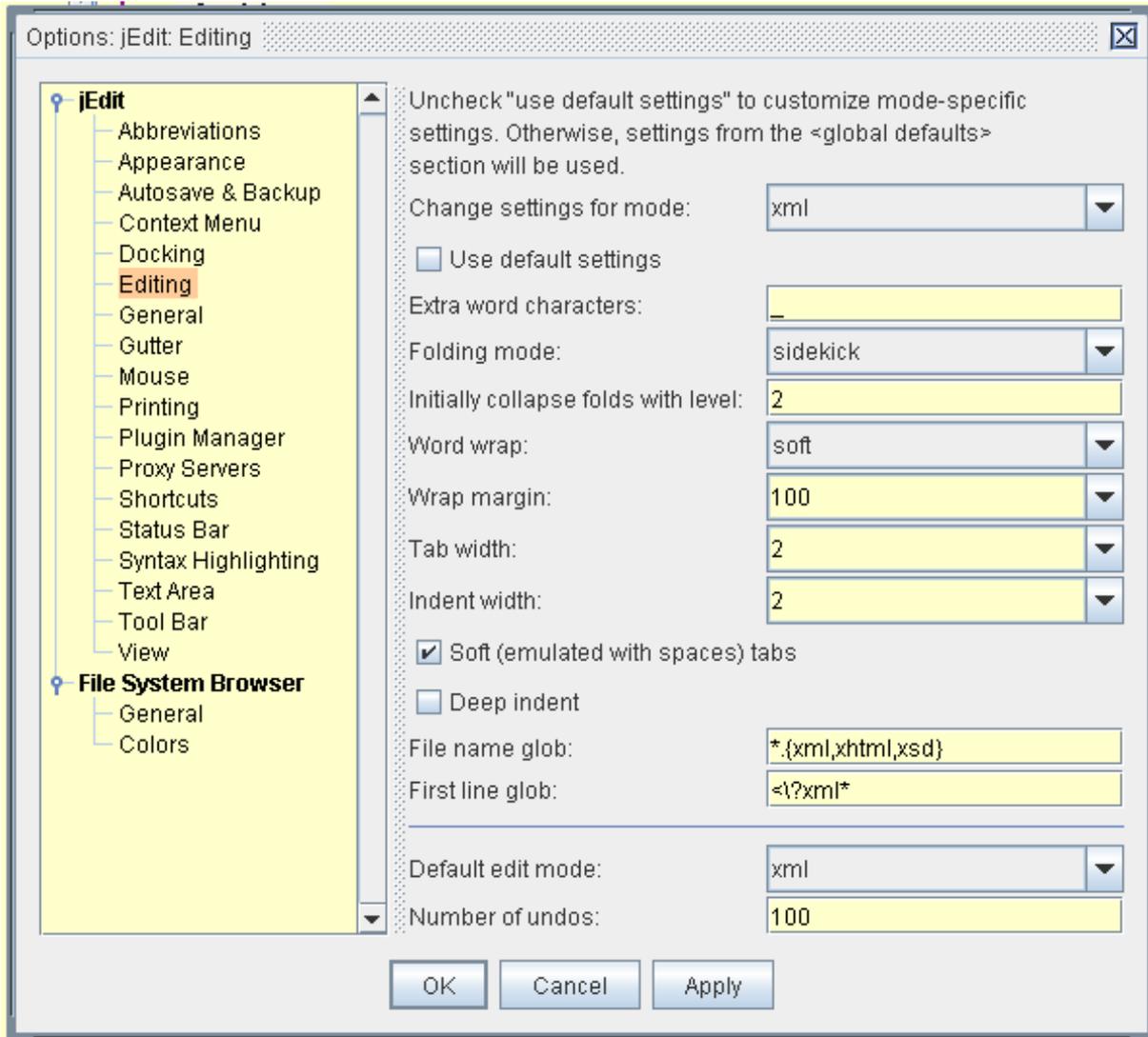


Figure 7.2. jEdit settings for editing options

7.2.4. A Sample View of jEdit

Figure 7.3, "A Sample View of jEdit" [23]

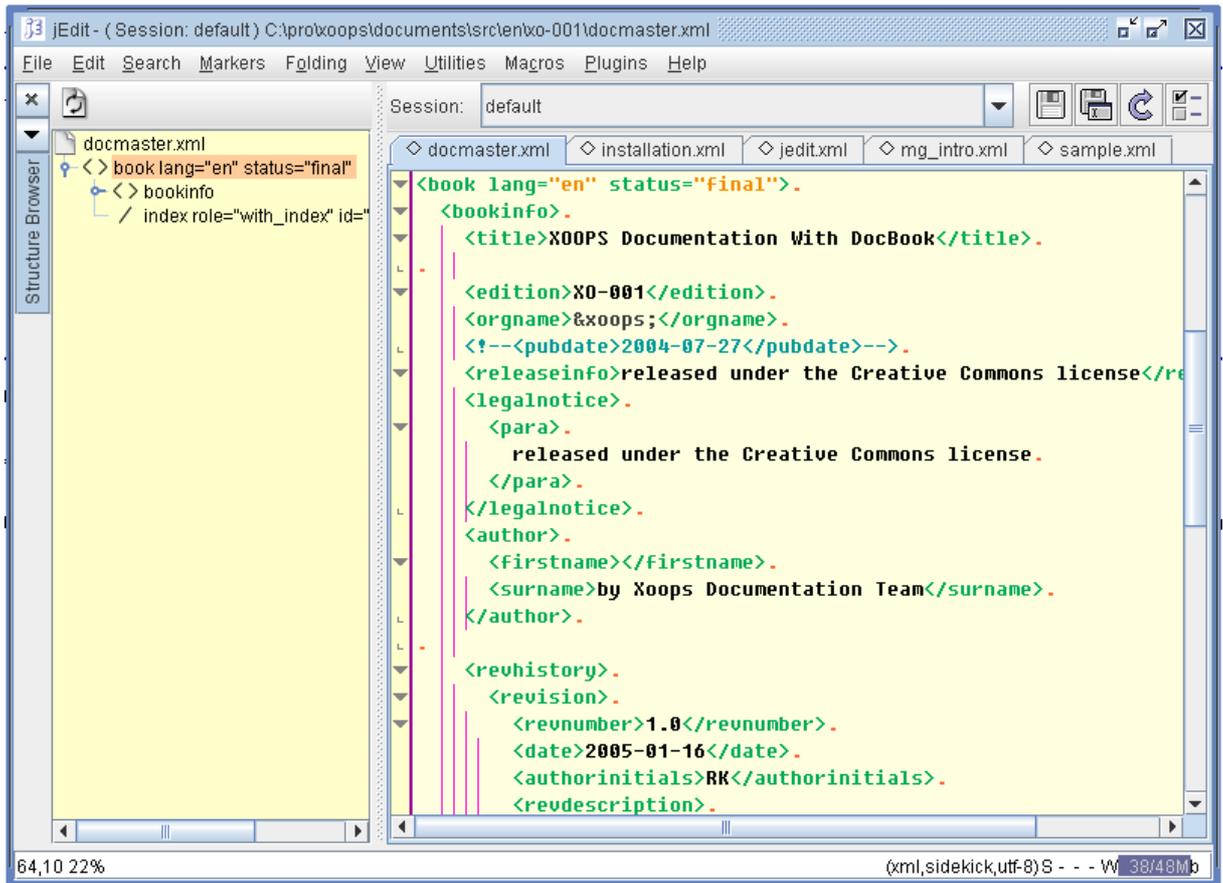


Figure 7.3. A Sample View of jEdit



Index

A

Animal
lion, 10
tiger, 12

C

CMS
Xoops, 5

D

DocBook, 5

T

table, 12