

DocBook HOWTO

Jorge Godoy

[Conectiva S.A.](#)

Publishing Department

godoy@conectiva.com

godoy@metalab.unc.edu

Revision History

Revision 1.3	February, 17th 2000	Revised by: godoy
Initial version.		
Revision 1.4	June, 24th 2000	Revised by: godoy
English corrections and added some more explanation on DocBook usage. I've added CSS to my stylesheets and if you have a CSS enabled browser you'll see a lot of difference from the previous version. I've corrected and added information based on all feedback I received since the first version of it.		
Revision 1.5	June, 26th 2000	Revised by: godoy
English corrections made by Horylka. Thanks.		

Table of Contents

1. [Introduction](#)
2. [Documents](#)
3. [Configuration needed](#)
4. [Creating and modifying catalogues](#)
5. [Tools for Document Edition Using DocBook](#)
6. [Writing with DocBook elements](#)
7. [Encoding Indexes](#)
8. [Inserting Pictures](#)
9. [Tables](#)
10. [Listings and program codes](#)
11. [Tools & Hints](#)
12. [Document samples](#)

[Glossary](#)

[Index](#)

This document's sole purpose is to provide basic knowledge about DocBook and explain how to use DocBook to write several types of documents; not all of DocBook's functions aren't explained.

This document will emphasize how to create books, guides, and articles that will be published over the Internet.

Any suggestions, corrections and comments regarding this document can be sent to the author: [Jorge Godoy](#).

An original copy of this document can be obtained by the copy of the files [using-docbook.sgml](#), [example-book.sgml](#), [example-article.sgml](#), [example-table.sgml](#), [compiles-sgml.sgml](#), [dsl-sample.sgml](#) and [glossary.sgml](#).

You can find this document as a one page HTML file [here](#). This should be easy downloading and/or printing this document.

1. Introduction

DocBook defines a set of markup elements useful for marking up text so that the text can then be transformed into several different formats.

It's possible to create documents in different formats HTML, XML, RTF, TeX, and others.

The idea is to write just once and reach the largest possible number of people with the information.

Digital information not stored properly tends to get lost. Due to the fact that not containing uncommon characters (such as binary formats) it's possible to index and search directly on the documents written on SGML and consequently on DocBook.

The SGML systems use markups to make their description. DocBook holds over 300 markup elements each one with several attributes which can assume several values, these can be fixed or defined by the document / style that the author has used.

Just to remind if any changes are made on the DocBook's definitions DTD, it's no longer DocBook.

2. Documents

This document uses the following conventions^[1]:

Descriptions

Warnings

Appearance

Caution
Warnings.

Descriptions

Hint

Notes

Informations requiring special attention

Appearance

Tip: Hint.

Note: Note.

Warning
Warning.

File Names

file.extension

Directory Names

directory

Commands to be typed

command

Applications Names

application

Prompt of users command

\$

Prompt of root users command

#

Environment Variables

VARIABLE

Emphasized work

word

Code Example

<para>Beginning and end of paragraph</para>

3. Configuration needed

The identifier systems used by the SGML and by some tools are based on catalogues which perform the translation of these identifiers over to files holding the necessary definitions.

The section on tailoring a catalogue (see [Section 4](#)) will give more details about these files.

For such tools to be able to find the necessary catalogue(s) the value of the environment variable SGML_CATALOG_FILES should be configured, as shown in the following example:

```
$ export SGML_CATALOG_FILES="/usr/lib/sgml/catalog"
```

This is the only necessary additional configuration for the DocBook, tools and the like to work correctly on your platform.

4. Creating and modifying catalogues

A catalogue is a text file containing the translation rules of the public identifier to system's files.

They make easy to use the DocBook, for they allow each user to have their files installed in a different place (e.g. your home directory, `/usr/local/sgml` or in any other place) though no other change on the document is necessary to be processed and "compiled".

Example 1. Example of catalogue

```
-- Catalogue for the Conectiva Styles -- ❶  
  
OVERRIDE YES  
  
PUBLIC "-//Conectiva SA//DTD DocBook Conectiva variant V1.0//EN" ❷  
    "/home/ldp/styles/books.dtd"  
  
DELEGATE "-//OASIS"  
    "/home/ldp/SGML/dtds/catalog.dtd"  
  
DOCTYPE BOOK /home/ldp/SGML/dtds/docbook/db31/docbook.dtd ❸  
  
-- EOF --
```

❶

Comment. Comments start with "--" and follow to the end of the line.

❷

The public type association `"//Conectiva SA//DTD books V1.0//EN"` with the file `books.dtd` on the directory `/home/ldp/styles`.

❸

Comment informing the end of the file.

As in the example above, to associate an identifier to a file just follow the sequence shown:

1. Copy the identifier *PUBLIC*
2. Type the identifying text
3. Indicate the path to the associated file

4.1. Explaining the terminology system

Notice the identifier

```
"//Conectiva SA//DTD books V1.0//EN"
```

Its formation is not random and follows some pre-defined conditions.

The token "-" indicates that the used identifier isn't a registered type. Only a few identifiers are registered and those usually belong to entities like ISO, IEEE, and others.

The second part of the identifier defines the name of the organization that created it. On the example above, Conectiva S.A.

The one before the last defines the contents (in this case a DTD[2]) and the name of the identified text.

The last element indicates the language in which the document was written. Since DocBook is a DTD written in English, the language is EN. The two letter code recommended is the ISO identification of the language.

More information can be obtained at [OASIS Technical Resolution 9401:1997 \(Amendment 2 to TR 9401\)](#).

4.2. Useful command for catalogues

The most common commands to be used on catalogues are:

PUBLIC

The keyword PUBLIC maps public identifiers for identifiers on the system.

SYSTEM

The keyword SYSTEM maps system identifiers for files on the system.

```
SYSTEM "http://nexus.conectiva/utilidades/publicacoes/livros.dtd"
"publicacoes/livros.dtd"
```

SGMLDECL

The keyword SGMLDECL designates the system identifier of the SGML statement that should be used.

```
SGMLDECL "publishings/books.dcl"
```

DTDDECL

Similar to the SGMLDECL the keyword DTDDECL identifies the SGML statement that should be used. DTDDECL makes the association of the statement with a public identifier to a DTD. Unfortunately this association isn't supported by the charge free tools available. The benefits of this statement can be achieved somehow with multiple catalogue files.

```
DTDDECL "-//Conectiva SA//DTD livros V1.0//EN" "publicacoes/livros.dcl"
```

CATALOG

The keyword `CATALOG` allows a catalogue to be included inside another. This is a way to make use of several different catalogues without the need to alter them.

OVERRIDE

The keyword `OVERRIDE` informs whether an identifier has priority over a system identifier. The standard on most systems is that the system identifier has priority over the public one.

DELEGATE

The keyword `DELEGATE` allows the association of a catalogue to a specific type of public identifier. The clause `DELEGATE` is very similar to the `CATALOG`, except by the fact that it doesn't do anything until a specific pattern is specified.

DOCTYPE

In case of a document starts with a type of document, but has no public identifier and no system identifier the clause `DOCTYPE` makes the association of this document with an specific DTD.

5. Tools for Document Edition Using DocBook

Several tools are available to facilitate the edition work of documents on the DocBook format. Due to the complexity of its rules and the large variety of element options and attributes to them, the use of these automatic tools are recommended.

A comprehensive list can be found at [the OASIS tools list](#).

Particularly and without the intention to start any religious "war" among editors, I think the solution emacs + psgml is the best solution at hands, regarding either the open source or the free of charge issue.

The tool used is according to each author's preference, however, it's recommended to have at least the following features:

- Analysis of the DTD dynamically
 - Insertion of elements based on DTD analyzed
 - Visual highlight of the text either in color or any other possible attributes. This helps to differentiate the text from constant processing elements and instructions on it.
-

6. Writing with DocBook elements

An editor capable of inserting an element according with DTD analisys helps a lot since it can allow or not the element to be used at the position where the cursor is in. This way you can be sure that no invalid element was added anywhere in your document.

In order to ensure future changes are as easy as possible, authors should try to keep as much compatibility as possible with theXML version of the DocBook DTD. This means keeping element names in upper case, using double quotes in all attributes, not using "markup minimizations" (explained below), and not omitting end tags. Most tools that automatically insert elements (like psgml+emacs) follow these rules automatically or with some fine tuning.

There are several forms of markup minimization. These include empty tags. One example of tag minimization is that instead of typing the end tag you simply type `</>`. Another example, as said before, is ommiting tags. You can see both examples below:

```
<para>I'm using <emphasis>here</>, normal text here,  
and <here</> I emphasized the text again, with empty tags.</para>
```

Each type of document created has a specific structure and example of documents can found afterwards on this document. (see [Section 12](#)).

Considering the explanation above we can proceed to instructions on how to write a document using DocBook.

6.1. Useful commands

The [Table 1](#) shows some commands which are useful to generate generic documents. Remember that some elements are valide only on some contexts.

Tip: Sometimes the appearance of a particular tag changes from one format to another. As a beginner in DocBook writing, you ay wish to see how your document looks in several formats before you publish them.

Note: Since the formatting depends on the output style chosen, it's recommended to use as much markup as possible. Even if the appearance of the output doesn't seem to change with the standard output style, there may be specific output formats that will make these tags stand out.

Table 1. Useful commands

Description	Command	Result
-------------	---------	--------

Description	Command	Result
E-mail address	<code><email>address@domain</email></code>	<code><address@domain></code>
About the author	<code><author>...</author></code>	(see example below)
Author's name	<code><firstname>First_Name</firstname></code> <code><othername>Middle_Name</othername></code> <code><surname>Surname</surname></code>	First Name Middle Name Surname
Keys' name (printings on the keyboard)	<code><keycap>F1</keycap></code>	F1
Symbol represented by the keys	<code><keysym>KEY_F1</keysym></code>	KEY_F1
Key's code	<code><keycode>0x3B</keycode></code>	0x3B
Combinations or sequences of keys	<code><keycombo></code> <code> <keycap>Ctrl</keycap></code> <code> <keycap>S</keycap></code> <code></keycombo></code>	Ctrl-S
Programs Menu	<code><guimenu>File</guimenu></code>	File
Menu Items	<code><guimenuitem>Salvar</guimenuitem></code>	Save
Menu Sequences	<code><menuchoice></code> <code> <shortcut></code> <code> <keycombo></code> <code> <keycap>Ctrl</keycap></code> <code> <keycap>S</keycap></code> <code> </keycombo></code> <code> </shortcut></code> <code> <guimenu>Arquivo</guimenu></code> <code> <guimenuitem>Salvar</guimenuitem></code> <code></menuchoice></code>	File->Save (Ctrl-S)
Mouse Button	<code><mousebutton>left</mousebutton></code>	left
Command Names	<code><command>comando</command></code>	command
Application Names	<code><application>application</application></code>	application
Text Bibliographical Reference	<code><citation>reference</citation></code>	[reference]

Description	Command	Result
Quote	<pre><blockquote> <attribution>Text Author</attribution> <para>Quote Text.</para> </blockquote></pre>	Quote Text. --Text Author
Index	(NA)	See Section 7 .
File Names	<pre><filename>file</filename></pre>	file
Directories	<pre><filename id="directory">directory</filename></pre>	directory/
Emphasize Text [a]	<pre><emphasis>text</emphasis></pre>	<i>text</i>
Footnotes	<pre><footnote> <to>Footnote text</to> </footnote></pre>	(See note at the end of this table)
URLs	<pre><ulink url="http://www.conectiva.com">Conectiva S.A.</></pre>	Conectiva S.A.
Markups List	<pre><itemizedlist> <listitem> <to>item</to> </listitem> <listitem> <to>item</to> </listitem> </itemizedlist></pre>	<ul style="list-style-type: none">• item• item
Numbered List	<pre><orderedlist> <listitem> <to>item</to> </listitem> <listitem> <to>item</to> </listitem> </orderedlist></pre>	<ol style="list-style-type: none">1. item2. item
Segmented	<pre><segmentedlist> <title>Binary to decimal conversion</title> <segtitle>Binary</segtitle> <segtitle>Decimal</segtitle> <seglistitem><seg>00</seg><seg>0</seg> </seglistitem> <seglistitem><seg>01</seg><seg>1</seg> </seglistitem> <seglistitem><seg>10</seg><seg>2</seg> </seglistitem> </segmentedlist></pre>	Binary to Decimal Conversion Binary: 00 Decimal: 0 Binary: 01 Decimal: 1 Binary: 10 Decimal: 2

The text can be emphasized in a few ways. The most common ways are italics and bold. DocBook, however, supports only italics. The use of bold requires additional settings on the stylesheet used.

7. Encoding Indexes

The generation of indexes depends on the markups inserted in the text. Such markups will be processed afterwards by an external tool and will generate the index. An example of such a tool is the collateindex.pl script (see [Section 11.1](#)). Details about the process used to generate these indexes are shown in [Section 11.3](#).

The indexes have nesting levels. The markup of an index is done by the code [Example 2](#).

Example 2. Code for the generation of an index

```
<indexterm>
  <primary>Main level</primary>
  <secondary>Second level</secondary>
  <tertiary>Third level</tertiary>
</indexterm>
```

It's possible to refer to chapters, sections and other parts of the document using the *attribute* zone.

Example 3. Use of the attributte zone

```
<section id="encoding-index">
  <title>Encoding Indexes</title>

  <indexterm zone="encoding-index">
    <primary>edition</primary>
    <secondary>index</secondary>
  </indexterm>

  <para>The generation of indexes depend on the inserted markups on the text. </para>
```

The [Example 3](#) has the code used to generate the entry of this edition on the index. In fact, since the attribute zone is used, the index statement could be located anywhere in the document or even in a separate file.

However, to facilitate maintenance the entries for the index were all placed after the text to which it refers.

Example 4. Usage of values startofrangeand endofrange on the attributeclass

```
<PARA>Typing the text normally sometimes there's the need of
<INDEXTERM CLASS="startofrange" ID="example-band-index">
  <PRIMARY>examples</PRIMARY>
  <SECONDARY>index</SECONDARY>
</INDEXTERM>
mark large amounts of text.</para>

<para>Keep inserting the paragraphs normally.</para>

<para>Until the end of the section intended
to be indexed.
<INDEXTERM STARTREF="example-band-index" CLASS="endofrange">.
</PARA>
```

8. Inserting Pictures

It's necessary to insert pictures for all types of media on which the document will be published.

If you use the TeX format you'll need the images as a PostScript file. For online publishing you can use any kind of common image file, such as JPG, GIF or PNG.

The easiest way to insert pictures is the use of the attribute `fileref`. Usually pictures are generated in JPG and in PostScript (PS or EPS).

Example 5. Inserting a picture

```
<figure>
  <title>Picture's Title</title>
  <graphic fileref="images/file"></graphic>
</figure>
```

Replacing `<figure>` by `<informalfigure>` eliminates the need to insert a title for the picture.

There's still the `float` attribute on which the value `0` indicates that the picture should be placed exactly where the text flux appears. The value `1` allows the picture to be moved to a more convenient location (this location can be described on the style sheet used or even can be controlled by the application being used).

8.1. Alternative Methods

The first alternative to [Example 5](#) is the elimination of elements `<figure>` or `<informalfigure>`.

Another interesting alternative when it's the decision to publish the text on media and pictures aren't accepted, is the use of a wrapper `<imageobject>`.

Example 6. Using `<imageobject>`

```
<figure>
  <title>Title</title>
  <mediaobject>
    <imageobject>
      <imagedata fileref="images/file.eps" format="eps">
    </imageobject>
    <imageobject>
      <imagedata fileref="images/file.jpg" format="jpg">
    </imageobject>
    <textobject>
      <phrase>Here there's an image of this example</phrase>
    </textobject>
    <caption><para>Image Description. Optional. </para></caption>
  </mediaobject>
</figure>
```

Files on the following formats are available BMP, CGM-BINARY, CGM-CHAR, CGM-CLEAR, DITROFF, DVI, EPS, EQN, FAX, GIF, GIF87A, GIF89A, IGES, JPEG, JPG, LINESPECIFIC, PCX, PIC, PS, SGML, TBL, TEX, TIFF, WMF, WPG.

This method presents an advantage: a better control of the application. The elements `<imageobject>` are consecutively tested until one of them is accepted. In case of the output format doesn't support images the element `<textobject>` will be used. However, the biggest advantage in usage of the format [Example 6](#) is that on the release 5.0 of the DocBook the element `<graphic>` will cease to exist.

As a disadvantage there's the need for more than one representation code of the same information. It's up to the author to decide which method he will implement illustrations and pictures on his or hers document, but for compatibility reasons with future versions *I recommend* the use of this method for pictures and graphics.

9. Tables

Many information are best represented when formatted as tables.

A primitive way to create tables was already presented on the [Table 1](#) with the use of `<simplelist>`, however, the DocBook has more sophisticated methods to deal with this information.

Example 7. Inserting tables

```
<table frame="all">
  <title>Sample Table</title>
  <tgroup cols="5">
    <colspec colname="column1">
    <colspec colname="column2">
    <colspec colname="column3">
    <colspec colnum="5" colname="column5">
    <spansec namest="column1" nameend="column2" spanname="span-horiz" align="center">
    <spansec namest="column2" nameend="column3" spanname="span-horiz-vert" align="center">
  <thead>
    <row>
      <entry spanname="span-horiz">
        <foreignphrase>Span</foreignphrase> horizontal
      </entry>
      <entry>Heading 2</entry>
      <entry>Heading 3</entry>
      <entry>Heading 4</entry>
    </row>
  </thead>
  <tfoot>
    <row>
      <entry>Footing 1</entry>
      <entry>Footing 2</entry>
      <entry>Footing 3</entry>
      <entry>Footing 4</entry>
      <entry>Footing 5</entry>
    </row>
  </tfoot>
</table>
```

```
</row>
</tfoot>
<tbody>
  <row>
    <entry>Data11</entry>
    <entry>Data12</entry>
    <entry>Data13</entry>
    <entry>Data14</entry>
    <entry>Data15</entry>
  </row>
  <row>
    <entry>Data21</entry>
    <entry>Data22</entry>
    <entry>Data23</entry>
    <entry>Data24</entry>
    <entry morerows="1" valign="middle">
      <foreignphrase>Span</foreignphrase> vertical
    </entry>
  </row>
  <row>
    <entry>Data31</entry>
    <entry spanname="span-horiz-vert" morerows="1" valign="bottom">
      <foreignphrase>Span</foreignphrase> duplo
    </entry>
    <entry>Data34</entry>
  </row>
  <row>
    <entry>Data41</entry>
    <entry>Data44</entry>
    <entry>Data45</entry>
  </row>
</tbody>
</tgroup>
</table>
```

Table 2. Example Table

Horizontal Span		Heading 2	Heading 3	Heading 4
Data11	Data12	Data13	Data14	Data15
Data21	Data22	Data23	Data24	Vertical Span
Data31	Double Span		Data34	
Data41			Data44	
Footing 1	Footing 2	Footing 3	Footing 4	Footing 5

10. Listings and program codes

An interesting feature is to show parts of code and the possibility to comment on them. The DocBook allows the insertion of the program code and also callouts to specific lines of this code.

Such a feature was used, for example, to demonstrate how a catalogue file is configured (see [Section 4](#)).

The used code was demonstrated below. In case the callout feature isn't desired, it's possible to eliminate the areas between `<areaspec>` and `<calloutlist>`.

```
<example id="sample-catalog">
  <title>Catalog Sample</title>
  <programlistingco>
    <areaspec>
      <area coords="1" id="ex.catalogue.comment">
        <area coords="5" id="ex.catalogue.definition">
          <area coords="11" id="ex.catalogue.eof">
            </areaspec>
            <programlisting>
-- Catalogues for the Conectiva S.A. Style --

OVERRIDE YES

PUBLIC "-//Conectiva SA//DTD books V1.0//EN" "/home/ldp/estilos/livros.dtd"

DELEGATE "-//OASIS" "/home/ldp/SGML/dtds/catalog.dtd"

DOCTYPE BOOK /home/ldp/SGML/dtds/docbook/db31/docbook.dtd

-- EOF --
      </programlisting>
    <calloutlist>
      <callout arearefs="ex.catalogue.comment">
        <to> Comment. Comments begin with <quote>--</quote>
          and follows to the end of the line. </to>
      </callout>
      <callout arearefs="ex.catalogue.definition">
        <to> The public type association
          <parameter class="option">"-//Conectiva SA//DTD books V1.0//EN"</parameter>
            with the file <filename>books.dtd</filename> on the directory
            <filename class="directory">/home/ldp/estilos</filename>. </para>
        </callout>
      <callout arearefs="ex.catalogue.eof">
        <para> Comment informing the end of the file. </para>
      </callout>
    </calloutlist>
  </programlistingco>
</example>
```

The listings can be directly inserted on the document's body without the need of the element `<example>` or `<para>`.

The calling coordinates specifications are done with reference to the code line which will be commented.

11. Tools & Hints

The process of producing output and generating indexes is repetitive and error prone. To make things easier some scripts are included here to facilitate this work. Customize and use them at will.

11.1. Compiling the sources

The script `compiles-sgml` is a set of grouped commands. As parameters the name of the document should be passed *SGML* and the output format expected.

The script version included below supports the formats *XML*, *HTML*, *TeX*, *RTF*, *PS*, *DVI* and mirrored PS, ideal for the creation of photolithographs.

The generation of the indices is made automatically by the script `collateindex.pl`[\[4\]](#), which should be installed in your system.

Besides the commands below which generate the outputs in different formats, there are other tools from Cygnus™ making the direct conversion. Such tools can be obtained in [here](#).

The list below is available [here](#).

Here is also available a version of [collateindex.pl](#).

Example 8. Script `compiles-sgml`

```
#!/bin/bash
#
# Compile DocBook documents into several output formats.
#
# Godoy.
# 19991230 - Initial release.
# 20000117 - Placed the options using "case" and parameters passed
#           via command line. The pages on the Zope are already updated.
#           --- Removed to public version (/home/ldp).
# 20000120 - Placed the call to use the books.dtd.
# 20000126 - Placed the commands for the index generation.
#
# If the jade is already installed, disconsider the line bellow.
JADE=/usr/bin/jade
# If the jade package is already installed, disconsider the line bellow.
# JADE=/usr/bin/openjade
DOCUMENT=$1
shift 1
TYPE=$1
```

```

. ~/.bash_profile
. ~/.bashrc

case $TYPE in
    html)
        rm -f *.htm
        rm -f *.html
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -N -o index.sgml
        jade -t sgml -V html-index -d /home/ldp/SGML/style/dsssl/docbook/html/docbook.dsl $DOCUMENT.sg
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -o index.sgml HTML.index
        $JADE -t sgml -i html -d /home/ldp/SGML/style/dsssl/docbook/html/docbook.dsl -d /home/ldp/SGML
        ;;
    rtf)
        rm -f $DOCUMENT.rtf
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -N -o index.sgml
        jade -t sgml -V html-index -d /home/ldp/SGML/style/dsssl/docbook/html/docbook.dsl $DOCUMENT.sg
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -o indice.sgml HTML.index
        $JADE -t rtf -V rtf-backend -d /home/ldp/SGML/style/dsssl/docbook/print/docbook.dsl -d /home/l
        ;;
    xml)
        rm -f $DOCUMENT.xml
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -N -o index.sgml
        jade -t sgml -V html-index -d /home/ldp/SGML/style/dsssl/docbook/html/docbook.dsl $DOCUMENT.sg
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -o indice.sgml HTML.index
        $JADE -t sgml -i xml -d /home/ldp/SGML/style/xsl/docbook/html/docbook.xsl $DOCUMENT.sgml
        ;;
    tex)
        rm -f $DOCUMENT.tex
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -N -o indice.sgml
        jade -t sgml -V html-index -d /home/ldp/SGML/style/dsssl/docbook/html/docbook.dsl $DOCUMENT.sg
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -o indice.sgml HTML.index
        $JADE -t tex -V tex-backend -d /home/ldp/SGML/style/dsssl/docbook/print/docbook.dsl -d /home/l
        ;;
    dvi)
        rm -f $DOCUMENT.tex
        rm -f $DOCUMENT.dvi
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -N -o indice.sgml
        jade -t sgml -V html-index -d /home/ldp/SGML/style/dsssl/docbook/html/docbook.dsl $DOCUMENT.sg
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -o indice.sgml HTML.index
        $JADE -t tex -V tex-backend -d /home/ldp/SGML/style/dsssl/docbook/print/docbook.dsl -d /home/l
        jadetex $DOCUMENT.tex
        ;;
    mirror)
        rm -f $DOCUMENT.tex
        rm -f $DOCUMENT.dvi
        rm -f $DOCUMENT.mirror.ps
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -N -o indice.sgml
        jade -t sgml -V html-index -d /home/ldp/SGML/style/dsssl/docbook/html/docbook.dsl $DOCUMENT.sg
        perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -o indice.sgml HTML.index
        $JADE -t tex -V tex-backend -d /home/ldp/SGML/style/dsssl/docbook/print/docbook.dsl -d /home/l
        jadetex $DOCUMENT.tex
        dvips -h /home/ldp/estilos/skel/mirr.hd -O 1.5cm,3cm -f $DOCUMENT.dvi -o $DOCUMENT.mirror.ps
        ;;
    ps)
        rm -f $DOCUMENT.tex
        rm -f $DOCUMENT.dvi

```



```

rm -f $DOCUMENT.ps
perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -N -o indice.sgml
jade -t sgml -V html-index -d /home/ldp/SGML/style/dsssl/docbook/html/docbook.dsl $DOCUMENT.sg
perl /home/ldp/SGML/style/dsssl/docbook/bin/collateindex.pl -o indice.sgml HTML.index
$JADE -t tex -V tex-backend -d /home/ldp/SGML/style/dsssl/docbook/print/docbook.dsl -d /home/l
jadetex $DOCUMENT.tex
dvips -The 1.5cm,3cm -f $DOCUMENT.dvi -o $DOCUMENT.ps
;;
*)
echo "How to use: $0 file {html|tex|rtf|xml|ps|dvi|mirror}"
exit 1
esac

exit 0

```

Obviously such script can be modified forming a Makefile, optimizing some functions.

11.2. Inserting a summary on the initial articles page

A feature that might be interesting in some cases is the insertion of a summary on the initial page of an article. The DocBook articles does not include it as a standard feature.

To enable this it's necessary to make a modification on the stylesheet file used.

The example below describes the process and it's use is shown in [Example 8](#).

Example 9. Stylesheet to insert summaries in articles

```

<!DOCTYPE style-sheet PUBLIC "-//James Clark//DTD DSSSL Style Sheet//EN" [
<!entity html-docbook PUBLIC "-//Norman Walsh//DOCUMENT DocBook HTML Stylesheet//EN" CDATA DSSSL>
<!entity print-docbook PUBLIC "-//Norman Walsh//DOCUMENT DocBook Print Stylesheet//EN" CDATA DSSSL>
]>

<style-sheet>
<style-specification use="html">
<style-specification-body>

; Includes a summary at the beginning of an item.
(define %generate-article-toc% #t)

</style-specification-body>
</style-specification>
<style-specification use="print">
<style-specification-body>

; Includes a summary at the beginning of an item.
(define %generate-article-toc% #t)

</style-specification-body>
</style-specification>

```

```
<external-specification id="html" document="html-docbook">
<external-specification id="print" document="print-docbook">
</style-sheet>
```

11.3. Inserting indexes automatically

Although DocBook has markups for the composition of indexes, these are not generated automatically. The tool [collateindex.pl](#) allows the indexes to be generated automatically.

The way to use this script is described bellow and a practical example can be seen previously in this document (see [Example 8](#)).

1. Process the document with jade using the style to *HTML* with the option `-v html-index`.

```
$ jade -t sgml -d html/docbook.dsl -V html-index document.sgml
```

2. Generate the document `index.sgml` with [collateindex.pl](#).

```
$ perl collateindex.pl -o index.sgml HTML.index
```

For the above example to work, it's necessary to define an *external entity* by calling the file `index.sgml`.

Example 10. Configuring an external entity to include the index

```
<!doctype article PUBLIC "-//OASIS//DTD DocBook V3.1//EN" [
<!-- Insertion of the index -->
<!entity index SYSTEM "index.sgml">
]>
```

See also [Section 7](#) for information on how to insert necessary information on the text.

Note: Remember that if you're trying to get Table of Contents or Indexes on PS or PDF output you'll need to run `jadetex` or `pdfjadetex` at least three times. This is a TeX requirement and not a DocBook or related application one.

11.4. Making notes on the text while it's being written

An interesting feature while writing a text is to be able to check if such text will be presentable or not on the final draft. It's common to have several parts of the text

marked as drafts, especially when we're updating an already existent document.

DocBook allows along with an entity of parameters to include or not the insertions of specific parts of text in several places on the document based on the context. Sometimes for an upgrading we need to see how the document looks like or just have sketches of a new session or chapter, but we don't want this sketch to appear on the final draft.

With the use of parameter entities we can include or eliminate these drafts altering only one line at the beginning of a document.

Example 11. Use of parameter entities

```
<!entity % review "INCLUDE">
...
<![%review;[
<para>This paragraph will be included on the draft when the entity
"review" is defined with the value "INCLUDE". </para>
]]>
```

The entity `review` might have several texts defined as in [Example 11](#). When the changes on the text are considered final, it's necessary just to remove parts of the text relative to the 3rd. and 6th. lines.

To keep the draft definitions and don't show the text on the final draft, just alter the specification of the entity from `INCLUDE` to `IGNORE`.

11.5. Re-using parts of documents

An important characteristic about the *external entities* is about the re-using issue of text and documents.

With those characteristics it's possible to create files with text used several times (e.g. licenses and company policies) and simply include those files in the appropriate place.

An example and application of this characteristic was found previously in [Example 10](#). Another example is the *SGML* code of this HOWTO.

12. Document samples

As stated before each type of document has a particular heading and valid commands structure. The following sub-sections will provide heading and valid commands structures on articles and books.

These examples do not cover all possibilities and they are available here only with the intention to serve as generic guides for what it's possible to do.

12.1. Article example

```
<article class="whitepaper" id="using -docbook" lang="pt-br"><?dbhtml filename="using-docbook.html">

<artheader>
  <title>Como-Fazer DocBook</title>
  <author>
    <firstname>Jorge</firstname>
    <othername>Luiz</othername>
    <surname>Godoy</surname>
    <othername>Filho</othername>
    <affiliation>
      <orgname><ulink url="http://www.conectiva.com">Conectiva S.A.</ulink></orgname>
      <orgdiv>Publishing Department</orgdiv>
      <address><email>godoy@conectiva.com</email></address>
    </affiliation>
  </author>
  <revhistory>
    <revision>
      <revnumber>1.0</revnumber>
      <date>27 de janeiro de 2000</date>
      <authorinitials>godoy</authorinitials>
      <revremark>Versão inicial.</revremark>
    </revision>
  </revhistory>

  <legalnotice>
    <para>This document can be freely translated and distributed. It's released
      under the LDP License.</para>
  </legalnotice>

  <keywordset>
    <keyword>SGML</keyword>
    <keyword>DocBook</keyword>
    <keyword>DTD</keyword>
    <keyword>XML</keyword>
    <keyword>catalogs</keyword>
    <keyword>documents</keyword>
    <keyword>Publishing</keyword>
    <keyword>Conectiva</keyword>
    <keyword>configuration</keyword>
    <keyword>use</keyword>
    <keyword>tools</keyword>
    <keyword>HOWTO</keyword>
  </keywordset>

</artheader>
```

12.2. Book Example

```
<book id="network-systems-administration"><?dbhtml filename="html/networkadm.html">
  <bookinfo>
    <author>
      <firstname>Jorge</firstname>
      <surname>Godoy</surname>
      <affiliation>
        <orgname>&conectivasa;</orgname>
        <orgdiv>Publishing Department</orgdiv>
        <address role="email">godoy@conectiva.com.br</address>
      </affiliation>
    </author>
    <editor>
      <firstname>Jorge</firstname>
      <surname>Godoy</surname>
    </editor>
    <copyright>
      <year>2000</year>
      <holder>&conectiva;</holder>
    </copyright>
    <title>Network and Systems Administration Using Linux</title>
    <legalnotice>
      <para>The contents of this book can be freely used and distributed
        as far as the source is mentioned as a reference that is, its bibliograph.</para>
    </legalnotice>
    <edition>INTERNAL Edition for comments</edition>
    <publisher>
      <publishername>Conectiva S/A</publishername>
      <address><city>Curitiba</city><country>Brasil</country></address>
    </publisher>
    <revhistory>
      <revision>
        <revnumber>1.0</revnumber>
        <date>Maio de 2000</date>
        <revremark>First Edition</revremark>
      </revision>
    </revhistory>
    <isbn>ISBN#</isbn>
    <pubdate>$Data: 1999/12/30 $</pubdate>
  </bookinfo>
```

Glossary

attribute

One attributte makes available extra information regarding the element on which it appears. The attributes always appear as a name-value pair on the initialization pointers. Example of an attribute is *id="identification"*, which gives to the attribute *id* the value *identification*.

Document Type Definition (DTD)

Group of statements that define element names and their attributes specifying the rules for combinations and sequences. It's the DTD that define which elements can or cannot be inserted in the context on which the cursor is in.

DSSSL

DSSSL stands for Document Style Semantics and Specification Language. It's an ISO standard (ISO/IEC 10179:1996). The DSSSL standard is internationally used as a language for documents stylesheets pages for SGML.

element

The elements define the hierarchical structure of a document. The majority of elements have opening and closing pointers. Among these pointers, pieces of text or even the whole document being written can be found. There are empty elements which contains only opening pointers without any content.

entity

Entity is a name designated for some part of data so that it can be referenced by a name. These designations are made by a statement and the stored data might hold from simple characters to chapters or set of statements of a DTD. There are parameter entities generic, external, internal and of data on the SGML.

external entity

An external entity points to an external document. External entities are used to include texts on certain locations of a SGML document. Suggestions for its use includes sample screens, legal notes and chapters.

generic entities

An entity referenced by a name which starts with "&" and ends with semicolon is a generic entity. Most of the time these type of entities are used on the document and not on the DTD. There are two types of entities: external and internal which refers either to special characters or to text objects such as repeated sentences, names or chapters.

internal entity

An internal entity refers to part of the text and is often used as a shortcut for portions of a text frequently repeated.

parameter entity

An entity often used on the DTD. The entity's name starts with a percent sign (%) and it ends with a semicolon.

float

Objects such as side bars, pictures, tables and charts are called floats when they don't have a fixed placement on the text. For a printed text, the chart can appear either at the top or at the bottom of the page. It can also be placed on the next page if that's too large.

processing instruction

A processing instruction is a command passed to the document formatting tool. It starts with "<?". A sample of instruction is used on this document for the generated file's choice when the file is converted to HTML: `<?dbhtml filename="file.html">`

SGML

Standard Generalized Markup Language. It's also an international standard (ISO8879) which specifies rules for the creation of electronic documents in markup systems regardless the work platform used.

tag

An SGML element bounded by the marks "<" and ">". Tags are used to mark the semantic or the structure of a document. A sample is the tag `<title>` to mark the beginning of a title.

XML

eXtensible Markup Language. A subproduct of SGML created specifically for Internet use.

XSL

XML Style Language. XSL is to a XML document what a DSSSL style is for a SGML document. In fact, the style is a document XML.

Index

C

catalogue

creating

example, [Creating and modifying catalogues](#)

terminology, [Explaining the terminology system](#)

modifying, [Creating and modifying catalogues](#)

sample, [Creating and modifying catalogues](#)

- terminology, [Explaining the terminology system](#)
- catalogues
 - creating, [Creating and modifying catalogues](#)
- configurations, [Configuration needed](#)
- variables
 - SGML_CATALOG_FILES, [Configuration needed](#)
- conventions, [Documents](#)

D

- documents
 - re-use, [Re-using parts of documents](#)

E

- edition
 - examples, [Document samples](#)
 - article, [Article example](#)
 - book, [Book Example](#)
 - index, [Encoding Indexes](#)
 - tools, [Tools for Document Edition Using DocBook](#)
 - using DocBook, [Writing with DocBook elements](#)
 - commands, [Useful commands](#)
- entities
 - parameters
 - exemple, [Making notes on the text while it's being written](#)
 - usage, [Making notes on the text while it's being written](#)

F

- figures
 - inserting
 - figure, [Inserting Pictures](#)
 - mediaobject, [Alternative Methods](#)

G

- graphics
 - inserting
 - graphic, [Inserting Pictures](#)

mediaobject, [Alternative Methods](#)

L

listings

inserting, [Listings and program codes](#)

example, [Listings and program codes](#)

T

tables

inserting, [Tables](#)

example, [Tables](#)

tools

articles

summary, [Inserting a summary on the initial articles page](#)

compiling sources

compile-sgml, [Compiling the sources](#)

compiling the sources, [Compiling the sources](#)

indexes

automatic generation

See also edition, index

Notes

[1]

Please, take a look at the [source](#) to see how to get similar results on your documents. You should also remember that the way this appears to you depends on the format you're reading this document: online appearance is slightly different from the PostScript or PDF ones.

[2]

Here are valid: DTD, DOCUMENT, ELEMENTS, ENTITIES and NONSGML.

[4]

More information about indexes are available at [the page about index of Norman Walsh](#).