

Guidelines:Documentation_en

Please it's so important to follow the instructions below when documenting using DocBook.

Introduction

DocBook is a simply XML editor plus a DTD. Apart of the XML where we're going to edit our document, there're some style sheet which is actually which gives our document the final appearance. There're a couple of proceedings when working with DocBook

- Editar el XML.
- Edit the XML
- By using docbook2pdf, docbook2html, docbook2man you have to export the final document to PDF, HTML or just manpages

DocBook has got some advantajes over editing XML by hand, basically, DocBook has a lot of useful tags.

1. XML is basically ascii text, therefore we're allow to use our svn to manage all of our documentations. This is the key if we want to work as a real team,
2. It is able to export documention in PDF, HTML and man from the same source, It will let us to do not waste time.
3. It is a strongly language which make documentation in the same way everytime. Everybody is used to read Docbook documents.
4. It is a GNU Standar used in loads of projects

We already know that learning DocBook is not an easy process but it's a really worth effort. We have decided to use DocBook in any of our projects, so everyboyd must learn how to work with DocBook in order to work as a team, so then, everybody will be allow to edit any of the project's docs.

Here it is a small guide which will point out how to edit and work with DocBook, we'd suggest you to make some coffee and get ready to spend such a funny day 😊

Editing with DocBook

You can use whatever editor you want, as an advice, could be a great idea to use one which allows highligh syntax. This is the mail XML body:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V3.1//EN"
    "/usr/share/xml/docbook/schema/dtd/4.4/docbookx.dtd" >
<book>
  <chapter>
    <title>Capitulo 1</title>
    <sect1>
      <title>Seccion 1</title>
      <para>
        Erase una vez un lobo feroz que comía usuarios de Windows...
      </para>
    </sect1>
    <sect1>
      <title>Seccion 2</title>
      <para>
        Hace mucho, mucho tiempo, antes de que GNOME y KDE pusieran de
rodillas a CDE...
      </para>
    </sect1>
  </chapter>
</book>
```

This is a small example of a DocBook document. First off, we could noticed the XML heading, which defines the DTD (by using a file, which have to be in our system, and if not, use a correct path in order to get the correct file)

Next item is the tag one, `<book>`, this “says” we're going to start a book. There's another one `<article>` which works in a bit different way. We suggest you to copy&paste both and try to render the final document and see what's better.

Render with DocBook

Te harán falta los siguientes paquetes debian (es probable que me deje alguno): You will need this Debian packages (maybe we forget some...if so, sorry and feel free to email us with the correct solution)

```
docbook docbook-dsssl docbook-utils docbook-xml docbook-xsl openjade
```

El paso para “generar” la documentacion sera simplemente usar el comando con el XML donde has generado tu texto en formato DocBook/XML:

Next step is to generate the documentation, it is made with the same command you used with DocBook/XML document.

```
$ docbook2pdf test.xml
```

```
Using catalogs: /etc/sgml/catalog
Using stylesheet: /usr/share/docbook-utils/docbook-utils.dsl#print
Working on: /tmp/test.xml
```

Done .

Si has cometido algun error en la edición del texto te lo indicará. DocBook es MUY estricto. Si el comando ha tenido éxito tendras un PDF con tu texto. Puedes hacer la prueba con docbook2html o docbook2man ya que funcionan de una forma muy similar.

If you made a typo when editing the text, it would shows to you where. Keep in mind DocBook is so strict. If the command was run succesfully, you'd have a PDF with your final text. Try it with docbook2html or docbook2man, its work in a very close way.

Inserting conflictive text in a XML

Un problema típico a la hora de escribir XML es que el propio texto que queremos introducir DENTRO del XML contenga a su vez, tags similares a las del XML o incluso XML en si, por ejemplo, cuando queremos describir un ejemplo de uso. Para ello existe un tag especial que indica “este contenido se escribe tal cual, sin interpretar”. Este tag especial, se usa de la siguiente manera:

There's a tipical problems when editing using XML, for instance, if we want to use a text `_IN_` the XML which has similars tags or just XML in it owns. To overcome this, there's a special tag which works like: “this content is written as it is, do not process it”. This tag works like this:

```
<![CDATA[_TEXTO_]]>
```

Where `TEXTO` is the XML text, with tags or charatchers which are not allow such as: “&”, “>” ó “<”.

For instance:

```
<![CDATA[
  El tag   sirve en HTML para indicar negrita.
  El equivalente en DocBook es la cursiva, que es <emphasis></emphasis>
]]>
```

Listing items

Simple List

This is a simple list

```
<simplelist type='inline'>
  <member>A</member>
  <member>B</member>
  <member>C</member>
</simplelist>
```

List with bottoms or black spots

Esto seria una lista con puntos negros típica. El punto negro se puede cambiar por otros tipos de grafico. This is a typical black-spots-list. Black spots could be changed by other graphical-style

```

<para>
  La administración de Babel se divide en los siguientes apartados:
</para>
<itemizedlist>
  <listitem>
    <para>Usuarios de Babel</para>
  </listitem>
  <listitem>
    <para>Agentes</para>
  </listitem>
  <listitem>
    <para>Configuración del servidor</para>
  </listitem>
  <listitem>
    <para>Mantenimiento de la Base de Datos</para>
  </listitem>
</itemizedlist>

```

Numbered List

Igual que la anterior pero en vez de itemizedlist sera orderedlist. Si queremos cambiar el tipo de numero (para numeros romanos, p.e) hay que añadir el atributo numeration =“lowerroman” en el tag de orderedlist. Veamos el ejemplo:

This is like the list we see before, but rather than itemizedlist we will use orderedlist. If we want to change the number style (like romans numbers, for instance) we have to add numeration =“lowerroman” in the orderedlist tag. See the example below:

```

<para>
  Los principales contribuyentes a dicho proyecto han sido:
</para>
<orderedlist numeration="lowerroman">
  <listitem>
    <para>Hal Computer Systems y O'Reilly & Associates, de
      1991 a 1994.
    </para>
  </listitem>
  <listitem>
    <para>El grupo Davenport, de 1994 a 1998.</para>
  </listitem>
  <listitem>
    <para>El grupo <acronym>OASIS</acronym> de 1998 hasta hoy.</para>
  </listitem>
</orderedlist>

```

Sections, Chapters, sub-Sections and stuff

La estructura basica de un Book (libro) es diferente de la de un artículo, pero la gran mayoría de elementos son comunes. Vamos a describir la estructura básica:

The main body of a Book is quite different than an article one, anyway, mostly of the items are the same for both book and article. Let's see the main body:

- `<chapter>` - Book's chapter. This is the most important tag.
- `<title>` - Chapter's titles..
- `<para>` - Text Paragraphs. There's usually no problem if you want to set up a paragraph in the beginning of the chapter in order to write an introduction.
- `<sect1>` - First section. It should have a "title" tag inside of a "para".
- `<sect2>` - Second section. It should be included inside of the first section. It has got a "title" and a "para" and it could also have more subsections such as `sect3` or `sect4`

It's better if we take a look at this example provided on the Babel Enterprise documentation:

```
<chapter>
  <title>Administración de Babel</title>
  <para>La administración de babel se realiza desde la aplicación
Web.</para>
  <sect1>
    <title>Usuarios de Babel</title>
    <para>Algo que decir</para>
    <sect2>
      <title>Grupos de usuarios</title>
      <para>Mas que decir</para>
    </sect2>
  </sect1>
  <sect1>
    <title>Segundo título</title>
    <para>Más que decir</para>
  </sect1>
</chapter>
```

Macros & includes

Puedes importar un XML externo a modo de include con una sencilla sintaxis. Esto permite por ejemplo que secciones comunes de diferentes documentos compartan el mismo origen, por ejemplo, el apéndice de la licencia FDL. Basta con declarar en la cabecera una macro, definida por una palabra que luego será referenciada como:

You're allow to import an external XML using include sytanx. This will let you, for instance, to share the appendix between different documents. It's done just enough by inserting a macro on the document header, it will need to include a word which will be referenced with:

```
&macro;
```

An example:

XML header where we have to define the macro, in this example it will be called *include_fdl*

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V3.1//EN"
    "/usr/share/xml/docbook/schema/dtd/4.4/docbookx.dtd" [
<!ENTITY babel_version "v1.0">
<!ENTITY babel "Babel Enterprise">
<!ENTITY include_fdl SYSTEM "fdl.xml">
]>
```

Y para “ejecutar” esa macro, es decir, sustituir el texto, bastará con poner en el texto la llamada a la macro, esto es: To run this macro, that's to say, to replace the text it's simply using the macro, see:

```
&include_fdl;
```

Para utilizar macros de sustitucion de texto es igual de sencillo. En el ejemplo anterior se han declarado dos variables, 'babel_version' y 'babel' que seran sustituidas por su valor cuando en el texto se escriba

If you want to use macros as a replacement of a text it's either easy. On the example before, there's two variables which has been declared, 'babel_version' and 'babel' which will be replaced by its value when the text would be written

```
&babel_version;
```

How to use bold fonts in DocBook

Identify stylesheet you're using in your system. Docbook usually says stylesheet it's using in execution, so execute it, for example:

```
docbook2pdf babel.xml
Using catalogs: /etc/sgml/catalog
Using stylesheet: /usr/share/docbook-utils/docbook-utils.dsl#print
Working on: /datos/babel/trunk/babel_doc/en/babel.xml
```

Edit your stylesheet, in this case “/usr/share/docbook-utils/docbook-utils.dsl”, and add this code before the end, just below another “element” definition:

```
(element emphasis
  (if (equal? (attribute-string "role") "bold")
    (make sequence
      font-weight: 'bold
      (process-children))
    (make sequence
      font-posture: 'italic
      (process-children))
  )
)
```

Now you only need to use

```
<emphasis role='bold'>bold text</emphasis> in your Docbook code.
```

DocBook documentation's styling

Como se ha dicho ya en la introducción es importante que sigamos una serie de recomendaciones para hacer mas fácil el trabajo en grupo, entre las que podemos destacar:

It has been said already, but once again is not a pain at all. Well, it's so important to follow some rules in order to be succesfull when working as a team, so, these are some of the rules you should keep in mind when editing doc:

- Indent every child and it content. Example:

```
<chapter>
  <title>Administración de Babel</title>
  <para>
    La administración de babel se realiza desde la aplicación Web.
  </para>
  <sect1>
    <title>Usuarios de Babel</title>
    <para>
      Algo que decir
    </para>
    <sect2>
      <title>Grupos de usuarios</title>
      <para>
        Mas que decir
      </para>
    </sect2>
  </sect1>
  <sect1>
    <title>Segundo título</title>
    <para>
      Más que decir
    </para>
  </sect1>
</chapter>
```

- Paragraph tags' content (<para>) must be indented and in a new line. Lines width couldn't be longer than 80 columns, if it's necessary you should continue in the next line. **Do not trust on text editors.** Right example:

```
<para>
  Este sería un párrafo bastante largo cuyo ancho será superior a 80
  columnas. Si
  siguiesemos escribiendo en él, lo haríamos en una nueva linea, alineada
  al comienzo
  de la linea anterior. Este ejemplo es prueba de ello.
</para>
```

WRONG Example:

```
<para>Si este texto fuese para la documentación de los programas de Artica
no serviría porque está escrito en una línea muy larga y por tanto no sería
correcto. Además, el <para> debería de estar cerrado en la línea inferior.
</para>
```

*Every single child must be in a new line. This which follow is **WRONG**

```
<chapter><title>Titulo</title>
  <para>parrafo que va en este capitulo</para>
</chapter>
```

- La excepción a la regla anterior son los tags *inline*, es decir, aquellos que afectan a partes de un texto o párrafo. El siguiente tag `<acronym>` es correcto:
- There's one exception of the rule above, it's the *inline* tag, that's, those affecting a text part or a paragraph. This tag `<acronym>` is OK:

```
<para>El grupo <acronym>OASIS</acronym> de 1998 hasta hoy.</para>
```

- It's cool to separate tag's text in a medium and large blocks, for instance:

```
<para>
  big text block... gran bloque de texto... gran bloque de texto...
  gran bloque de texto... gran bloque de texto... gran bloque de texto...
  gran bloque de texto...
</para>
```

- Sé práctico escribiendo los párrafos, se prefieren varios párrafos cortos a uno largo y complicado.
- Be practical when writing paragraphs, it's preferred to use shorts paragraphs rather than a large and complicated ones
- Do not generate text using personals style-sheets, text must be generated using standars style-sheet, couse nobody is supposed to have YOUR personal style-sheet.

Important rules to keep in mind when writting documentation

Hay unas reglas de oro para que la documentación sea sencilla y fácil de leer. There's some important rules in order to make the documentation easy and simply to read.

Rule 1

- Each sentence should not be longer than 25 words

Example:

A sentence like this:

Bajo condiciones normales, el kernel no siempre escribe los datos a disco inmediatamente, si no que los guarda en un buffer de memoria para periódicamente escribirlos a disco y así agilizar las operaciones.

Should be re-write like this:

Generalmente, el kernel almacena el dato en memoria antes de escribirlo periódicamente a disco.

Rule 2

- Each paragraph is a subject
- Each sentence is an idea

Advice: You should plan paragraphs order before writing the document. You should, too, decide about what do you want to talk about in each paragraph

Rule 3

- Explicit examples should be used in order to show how something works. Write instructions instead of descriptions.

Example:

This text:

Hay una caja de texto que se puede usar para buscar la definición de una palabra.

It's better this way:

Para buscar la definición de una palabra, escribala en la caja de texto y pulse el botón "Buscar".

Rule 4

- Write in a neutral-style

Example:

La aplicación es un pequeño capturador de pantallas.

It's better this way:

Puede usar la aplicación para hacer una captura de pantalla.

References

Useful Documentation about DocBook

- DocBook. [Basic Manuals](#).
- DocBook. [Complete Manual](#).
- DocBook. [More manuals](#)
- [Short reference guide](#).
- [Style sheet reference guide](#)

From:

<https://pandorafms.com/manual/> - **Pandora FMS Documentation**

Permanent link:

https://pandorafms.com/manual/guidelines/documentation_en

Last update: **2021/11/05 12:05**

