

DocBook

DocBook - What is it?

- It's not a double-clickable application
- It's a way to structure technical, computer-oriented documentation

DocBook is Markup

- Adds Value
- Increases Cost
- For Example:

`<para>`

If `<function>open()</function>` was successful, the return value is a non-negative integer called a `<firstterm>`file descriptor`</firstterm>`, frequently shortened to just `<literal>fd</literal>`. This integer value is the index into a per-process table that references open files. This file descriptor is used as a handle to the file for subsequent calls.

`</para>`

So Where'd It Come From?

- AI to the rescue? Not quite
- Need to hold the computer's hand
- There's more than one market
- SGML

We Marksss its up, my preciousss..



SGML

- *Standard Generalized Markup Language*
- *A way to describe markup languages*
- *HTML is an SGML Application*
- *DocBook is an SGML Application*
- *DTDs*
- *XML*

XML

- "XML is basically the subset of SGML that Microsoft's developers could understand" -- Dan Lyke
- XML is an SGML Profile
- There is DocBook SGML and DocBook XML

DocBook

- Started in 1991
- HaL Computer Systems and O'Reilly
- Originally an interchange format
- OASIS and Norm Walsh
- Now at version 4.3
- Used by lots of organizations

But... Why?

- *Semantic Markup*
- *Tells what things are*
- *Not what they look like*

Doing More With Documents

- *Global formatting changes*
- *Intellegent searches*
- *Repurposing documents*
- *Driving processes from the document's structure*

DocBook! Woo Hoo!

- Tags delimit elements
- <start-tag>
- </end-tag>
- Elements can contain other elements
- DocBook says how you can structure things

DocBook Tags

abbrev	arg	bibliomisc
abstract	article	bibliomixed
accel	articleinfo	bibliomset
ackno	artpagenums	bibliorelation
acronym	attribution	biblioset
action	audiodata	bibliosource
address	audioobject	blockinfo
affiliation	author	blockquote
alt	authorblurb	book
anchor	authorgroup	bookinfo
answer	authorinitials	bridgehead
appendix	beginpage	callout
appendixinfo	bibliocoverage	calloutlist
application	bibliodiv	caption
area	bibliography	caution
areaset	bibliographyinfo	chapter
areaspec	biblioid	chapterinfo

DocBook Tags

citation	command	corpcredit
citebibliod	computeroutput	corpname
citerefentry	confdates	country
citetitle	confgroup	database
city	confnum	date
classname	confsponsor	dedication
classynopsis	conftitle	destructorsynopsis
classsynopsisinfo	constant	edition
cmdsynopsis	constraint	editor
co	constraintdef	email
code	constructorsynopsis	emphasis
col	contractnum	entry
colgroup	contractsponsor	entrytbl
collabl	contrib	envvar
collabname	copyright	epigraph
colophon	coref	equation
colspec	corpauthor	errorcode

DocBook Tags

errorname	funcprototype	guiicon	indexinfo
errortext	funcsynopsis	guilabel	indexterm
errortype	funcsynopsisinfo	guimenu	informalequation
example	function	guimenuitem	informalexample
exceptionname	glossary	guisubmenu	informalfigure
fax	glossaryinfo	hardware	informaltable
fieldsynopsis	glossdef	highlights	initializer
figure	glossdiv	holder	inlineequation
filename	glossentry	honorific	inlinegraphic
firstname	glosslist	html:form	inlinemediaobject
firstterm	glossee	imagedata	interface
footnote	glosseealso	imageobject	interfacename
footnoteref	glossterm	imageobjectco	invpartnumber
foreignphrase	graphic	important	isbn
formalpara	graphicco	index	issuenum
funcdef	group	indexdiv	itemizedlist
funcparams	guibutton	indexentry	itermset

Structural Tags

- book
- article
- section / sect[1-5]
- chapter
- appendix

Structural Tags in Action

```
<book>
  <bookinfo>
    <title>Core Mac OS X and Unix Programming</title>
  </bookinfo>
  ...
  <chapter id="files">
    <title>Files, Part 1: I/O and Permissions</title>
    ...
    <sect1>
      <title>Unbuffered I/O</title>
      ...
      <sect2>
        <title>Opening a file</title>
        ...
      </sect2>
    </sect1>
  </chapter>
</book>
```


Content Tags

- Block vs. Inline elements
- Formal vs. Informal elements
- quote
- firstterm
- para
- programlisting
- figure
- and on and on and on

Content Tags in action

`<para>`

The `<function>lseek()</function>` call is what is used to change the offset: `<synopsis>off_t <function>lseek</function>(int fd, off_t offset, int whence);</synopsis>` `<varname>offset</varname>`, in combination with `<varname>whence</varname>`, is used to locate a particular byte in the file. `<varname>whence</varname>` can have one of three values:

`</para>`

The `lseek()` call is what is used to change the offset:

```
off_t lseek (int fd, off_t offset, int whence);
```

`offset`, in combination with `whence`, is used to locate a particular byte in the file. `whence` can have one of three values:

Tag Soup

O_APPEND

Append on every write. This does an implicit seek to the end before writing, but is atomic.

O_NONBLOCK

Do not block for an open, and do not block when waiting for data.

O_CREAT

Create the file if it does not exist.

Tag Soup Is Good Food

```
<variablelist>
  <varlistentry>
    <term><literal>O_APPEND</literal></term>
    <listitem>
      <para>
        Append on every write. This does an
        implicit seek to the end before writing, but
        is atomic.
      </para>
    </listitem>
  </varlistentry>

  <varlistentry>
    <term><literal>O_NONBLOCK</literal></term>
    <listitem>
      <para>
        Do not block for an open, and do not block when
        waiting for data.
      </para>
    </listitem>
  </varlistentry>
  ...
</variablelist>
```

Attributes

```
<figure>
<title>File Descriptors</title>
<graphic fileref="files-chap/fds.pdf"></figure>
```

```
<chapter id="networking">
<title>Network Programming With Sockets</title>
```

The samples here will just call `<function>write()</function>` without a loop. `<xref linkend="networking">` (Networking) will demonstrate the paranoid way of calling `<function>write()</function>`.

Actually writing your masterpiece

- vi(m) and emacs
- OpenOffice.org
- LyX
- Commercial packages

I've Written My Book. Now What?

- The Chain Of Pain
- DocBook to:
- HTML
- PDF
- TeX
- DVI
- Postscript

Some Of The Tools

- *Get a pre-built package if you can*
- DocBook SGML or XML?
- Jade / DSSSL
- JadeTeX
- teTeX
- XSL / XSLT
- FOP

Should I Use It?

- DocBook is not easy!
- Is it worth your time?
- Can you use the document structure?

What about these "Other Applications"

- Exploiting the document's structure

CoreBook Structure

```
<CHAPTER>
<TITLE>
Files, Part 1: I/O and Permissions
</TITLE>

<SECT1>
<TITLE>
Unbuffered I/O
</TITLE>

<SECT2>
<TITLE>
Opening a file
</TITLE>

</SECT2>

<SECT2>
<TITLE>
Writing to a file
</TITLE>

...
```

Resources

- DocBook Books
- <http://docbook.org/>
- <http://wiki.docbook.org>
- <http://wiki.docbook.org/topic/DocBookAuthoringTools>
- [#docbook](http://irc.freenode.net) on irc.freenode.net