

Writing DocBook-XML pages for Deep Space 6

Simone Piunno
Deep Space 6

`simone@deepspace6.net`

Mauro Tortonesi
Deep Space 6

`mauro@deepspace6.net`

In this article we will explain how to write DocBook-XML pages for Deep Space 6. Authors are encouraged to use it as a reference and stylesheets designers can use it as a testbed for their sheets.

Table of Contents

The header.....	3
The article tag	3
My first section	3
Links and Lists	4
Programming	4
Tables and figures	4
Bibliography	5
Useful tools	5
References	6

The header

At the beginning of each DocBook-XML page you write, you must include the following header:

```
<?xml version="1.0"?>
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
    "/usr/share/sgml/docbook/xml-dtd-4.1.2/docbookx.dtd">
```

This header is needed to tell XML parsers that this document is a DocBook-XML file (an XML file that makes use of the above-specified DocBook DTD).

The article tag

After the header, you must open an article tag. Inside the article tag, you have to include both an articleinfo section and the actual content of the page.

The articleinfo tag

The purpose of the articleinfo tag is to contain the title of the page, the authors' name, an optional abstract and other similar information.

Here is a simple example:

```
<articleinfo>
  <title>The title of your page</title>
  <date>Publication date of the page (e.g. December 14th 2002)</date>
  <author>
    <firstname>Antonio</firstname>
    <surname>Cohimbra de la Coronilla y Azevedo</surname>
    <affiliation>
      <orgname>Duff Beer</orgname>
      <address><email>antonio@duffbeer.com</email></address>
    </affiliation>
  </author>
  <author>
    <firstname>Bastiano Joao</firstname>
    <surname>Cohimbra de la Coronilla y Azevedo</surname>
    <affiliation>
      <orgname>Duff Beer</orgname>
      <address><email>bastiano@duffbeer.com</email></address>
    </affiliation>
  </author>
  <abstract>
    <para>
      If you are writing an article, perhaps you are interested in writing
      an abstract, but this is absolutely optional.
    </para>
    <para>
      Remember that you can write abstracts that have more than one paragraph.
      Each paragraph must be included in a <para></para> tag couple.
    </para>
  </abstract>
</articleinfo>
```

My first section

In your article you will write many sections, each with its own title. *You don't need to number your sections*, because our XSL stylesheets will do it automatically for you. Note how we emphasize text.

Inside a section you can put as many paragraphs you want, it's easy as wrapping each paragraph with a `<para>` tag. You can even use subsections:

A sub section

Here you can put your subsection paragraphs

Another subsection

Isn't easy? Of course you can descend more than two levels, at most five, depending on the stylesheet.

Links and Lists

To link a portion of text with an URL, you can use the `<ulink>` tag.

Itemized lists can be written this way:

- one item
- another item

Maybe you don't want an itemized list, but instead a variable list:

1st term

explaining the 1st term

2nd term

explaining the 2nd term

There are many other list flavours, you can always refer to the official XML docbook specifications, v4.1.2, to check which tags are allowed in every part of your document.

Programming

Do you need to include program listings into your document? No problem!

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    printf("<?xml version='1.0'?>\n");
    printf("<stdout>Hello, world</stdout>\n");
    return (EXIT_SUCCESS);
}
```

Like you see, you can even use special characters if you pay attention to use `<![CDATA[` sections.

Tables and figures

Writing tables is not much different from what you are used to see in HTML, but here we have two kind of tables: `<table>` and `<informaltable>`. Both are compliant to the OASIS TM 9502:1995 standard but the former has a mandatory *title* and the latter hasn't it. Let's see an example:

Table 1. A long title for this short table

Some	text	carefully
interspersed	in our	cells.

Table can also serve to group **mediaobjects**.

Table 2. A mediaobject table



We can have images too! While we're at it let's explain that images are part of **mediaobjects** and can have a caption like this and multiple alternative representation (e.g. text, audio) to be used when the output device is not graphic-capable. The real image is referenced by filename.

`informaltables` are more or less the same, but without *title*

The	usual	text
interspersed	in our	cells.

Bibliography

Sometime you could want to refer to external documents and to group all the document pointers together in a **bibliography**. As you probably already see, this is very easy, you can just link the **biblioentry** this way: EXAMPLE. Cross-references are automatically calculated by docbook stylesheets.

Useful tools

While you write your document, there are some useful tools that can help you avoiding errors. The most important is an XML validator so that anytime you can make sure your file is compliant to the DocBook specification.

One of the easiest validators is **xmllint**, which you can find included in libxml2. You can validate this way:

```
xmllint --valid --noout test-article.xml
```

On a valid document you should receive no output. Any error will be reported like this:

```
$ xmllint --valid --noout test-article.xml
test-article.xml:225: validity error: Element para is not
      declared in para list of possible children
  </para>
    ^
```

References

[OASIS TM 9502:1995] *CALS Table Model Document Type Definition*¹, 1995, OASIS.

[EXAMPLE] *An example reference document*, 1789, Napoleon Bonaparte.

[DocBook] *DocBook: The Definitive Guide*², Norman Walsh, Leonard Muellner, 156592-580-7, 1999, 2000, 2001, 2002.

[libxml2] *libxml2 package*³.

Notes

1. <http://www.oasis-open.org/html/a502.htm>
2. <http://www.docbook.org/tdg>
3. <http://xmlsoft.org/>