

How to Write an Article with DocBook and DAPS

Tutorial

Vanessa Wallfahrer
Tanja Roth

DAPS is an open source program for transforming DocBook XML into output formats such as HTML or PDF. DAPS is command line based and runs on Linux*.

In XML-based publishing, layout and contents are strictly separated: while writing, you only take care of the contents and the XML tags to use. The layout for different output formats is defined in so-called stylesheets and is automatically added when converting the XML into output formats.

This tutorial is for DocBook and DAPS beginners. Previous knowledge of XML is not required, but you need basic knowledge of using the Bash Shell. We will write a recipe for a chili sauce in DocBook and convert it into HTML. The tutorial explains step-by-step how to set up your document, which basic XML tags to use for writing it, and how to produce the output you see in Figure 1.

Figure 1. The Goal: Publishing a Recipe for a Chili Sauce in HTML Format

Little G's Ceylon Bang

Tux Penguin

Table of Contents

[What do you need?](#)
[Ingredients](#)
[Equipment](#)
[Preparation](#)

Abstract

Recipe for a chili.

Before you start to cook, check out the following website: <http://www.crazyhotseeds.com/top-10-worlds-hottest-peppers/>

What do you need?

Ingredients

How to prepare see [the section called "Preparation"](#)

- 60g Habanero Chilis
- 30g Cayenne Chilis
- 1,5 Butch T Chilis
- 75g Kidney Beans


Equipment

- Hand blender
- Pot

Preparation

1. Rinse, then drain the kidney beans for about 10 minutes.
2. Fry kidney beans with some oil.
3. Puree all ingredients.
4. Cook for about 5 minutes.

Figure 1. Butch T Chili



Why Should I Use DocBook and DAPS?

DocBook is a markup language for technical documentation. Content is tagged with XML elements according to its meaning. This is similar to sorting objects into drawers according to their function: For example, you place scissors and tongs into a drawer labeled `Tools`, whereas you place teddies and

building blocks into a drawer labeled Toys. Similarly, when writing documents with DocBook, you would “sort” the author's name into an XML tag called `author`, whereas you would “sort” a reference to another document into an XML element called `xref`.

While this might seem cumbersome at first sight, it is a huge advantage when considering the different output formats you can generate from the same XML sources. Imagine you want anything tagged with an `xref` element to appear in *blue* in HTML output, whereas you want it to appear in *green* in the PDF output. Or you decide at one point that you want references to other documents (`xrefs`) to be displayed in red in all output formats from now on. This can be achieved *without* changing anything in the XML files themselves. It is only a matter of defining in the stylesheets which color to use for `xref` elements in different output formats.

With DAPS, you can convert your XML sources into various output formats with one command. DAPS takes care of converting any images in your documents automatically into the format best suited for the selected output format. DAPS also helps you in comfortably writing and editing your documents by providing additional features like editor macros, spell checking or link checking. With daps you can also control which stylesheets to use for generating output.

What You Need

Before you start writing, you need to install a few packages. To ease writing use an editor with XML support. Recommended is Emacs and DAPS version 2.0 or higher. Do the following to install Emacs and DAPS:

1. Open your terminal and log in as root by entering

```
tux:~> su -
```

Note that `tux:~>` or `(root # below)` stands for your prompt. Do not type that part.

2. Enter your root password.
3. Install Emacs with zypper:

```
root # zypper in emacs
```

4. Install DAPS with zypper:

```
root # zypper in daps
```

How to Start

To be able to process the XML files, DAPS requires a certain directory structure. Either use **daps-init** or set it up manually. The **daps-init** command is especially advisable for beginners, because it creates an example and the necessary structure. File names always end with `.xml`.

Working with daps-init

The **daps-init** command creates a working environment for DAPS. In this tutorial's example, it creates a directory called `tutorial`. It contains a DAPS configuration file for your tutorial (`DC-tutorial`), the DocBook XML file located in the `XML` subdirectory and a subdirectory containing image files (`images`). The `image` directory will be relevant later (see the section called “Integrating Images” [11]).

1. The **daps-init** command creates a working environment for DAPS. It builds a directory with your entered `filename`. In this directory are a `DC`-file, a `xml` subdirectory and a `image` subdirectory. The `image` directory (see the section called “Integrating Images” [11]) will be relevant later.

```
daps-init --docdir recipe -r article
```

2. Now it is your turn: write your text! To learn about the needed XML tags, read the following chapters. If you need help with DAPS, open the `daps-init` man page in your terminal:

```
tux:~> daps-init --help
```

3. Open the XML file (`recipe/xml/MAIN-daps-example.xml`) in your editor and use it as a reference.
4. Open the Documentation Configuration file (`recipe/DC-daps-example`) in an editor and use it as reference.
5. Proceed with the section called “How to Write”, where we will create another XML file and another DC file from scratch (in parallel to the files created by `daps-init`) and use them to write the recipe.

Working without daps-init

This method is for the more experienced writer.

1. For working without **daps-init**, create a directory with your project name and two subdirectories, called `xml` and `images`. Open a terminal.
2.

```
tux:~> mkdir recipe
```
3.

```
tux:~> cd recipe
```
4.

```
tux:~> mkdir -p images/src/{dia,eps,fig,pdf,png,svg} xml
```

Now you have created the required directory structure that is needed by DAPS.

5. Create a DC-file, which is a configuration file for your project. Open a new file and paste the following lines with your XML file name instead of “`MAIN-daps-example.xml`”. See below.

Creating an XML File and a Documentation Configuration File

Now we will create an XML file (that we will use in the following to write the recipe) and a Documentation Configuration (DC) file (that we will use later on to convert the XML file into HTML output).

1. **To create the XML file:**
 - a. In a terminal, switch to the XML subdirectory.

```
tux:~> cd xml
```
 - b. Open a new file, and paste the following header into it (see Example 1, “Header of a DocBook File”), or copy this from `MAIN-daps-example.xml`.

Example 1. Header of a DocBook File

```
<?xml version="1.0" encoding="utf-8"?> ❶  
<!DOCTYPE article PUBLIC ❷  
"-//OASIS//DTD DocBook XML V4.5//EN"  
"http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
```

]>

- ❶ XML-Declaration for XML V1.0, encoding UTF-8
- ❷ The general syntax for a document type declaration

Don't be afraid if you don't understand the header. Just insert it by copy and paste into your file.

- c. Save the file as `recipe.xml`.

2. To create the DC file:

- a. Switch to the directory above (that should be your `recipe` directory):

```
tux:~> cd ..
```

- b. Open a new file and paste the following line into it: `MAIN="recipe.xml"`
- c. Save the file, for example as `DC-recipe`.

How to Write

Identifying Start and End Tags

The most important thing you need to know about XML is, that you need to close the tags. Either with an end tag or with a backslash at the end of the tag (for example: `<para/>`). The content between the tags depends on which element you want to create. Open the XML file `recipe.xml` that you just created, and place the cursor below the header. To write an article, paste an **article** start and end tag into the XML file. Between these two tags, write your text as shown in Example 2, “Start- and Endtag of an Article”.

Example 2. Start- and Endtag of an Article

```
<article>
Tags and text
</article>
```

Adding Metadata

Each article may contain information about its author, publication date, release information, copyright or other “metadata”. To include such metadata use an article info element and enter the following data (see Example 3, “Article Info Header”):

Example 3. Article Info Header

```
<article lang="en" id="art.template">❶
<title>Little G's Ceylon Bang</title>❷
<articleinfo>
<releaseinfo>
  2013-10-21❸
</releaseinfo>
<author>❹
  <firstname>Tux</firstname>
  <surname>Penguin</surname>
```

```
</author>
</articleinfo>
<abstract>❶
Recipe for a chili sauce.
</abstract>
</article>❷
```

- ❶ Article start tag.
- ❷ Title for the article.
- ❸ Build date.
- ❹ The author's name.
- ❺ An abstract. What is the subject of your article?
- ❻ Article end tag. It always has to be at the end of the whole article.

Figure 2. HTML Output (Article Title and Article Info)



Structuring Your Document With Sections

Each article consists of sections and subsections. Example 4, “Example: Structure With Sections” contains a section and a subsection.

Example 4. Example: Structure With Sections

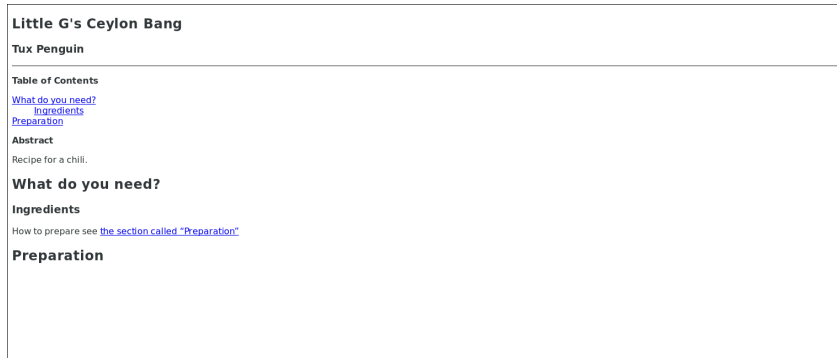
```
<sect1>❶
<title>What do you need?</title>
<sect2>❷
<title>Ingredients</title>
<para/>
</sect2>❸
<sect2>
<title>Equipment</title>
<para/>
</sect2>
</sect1>❹
<sect1 id="sec.preparation">
<title>Preparation</title>
</sect1>
```

- ❶ Start tag of section level one. Section1 is the parent element of every element that comes below. Always needs a title. Recommended is a para tag in every section.
- ❷ Start tag section of level two.
- ❸ End tag section of level two.

- 4 End tag of the section level one parent element.

Place the cursor behind the abstract end tag and in front of the article end tag and paste the content of “Example 4” into your XML file.

Figure 3. HTML Output (Article With Sections)



The image above shows the given XML file example as an HTML page. As you can see in the picture, the section level two is a child element of section level one. It is possible to nest sections.

Creating Itemized Lists

To create a listing, use the `<itemizedlist>` tag.

Example 5. Itemized List

```
<itemizedlist>
  <listitem>
    <para>
      60g Habanero Chilis
    </para>
  </listitem>
  <listitem>
    <para>
      30g Cayenne Chilis
    </para>
  </listitem>
  <listitem>
    <para>
      1,5 Butch T Chilis
    </para>
  </listitem>
  <listitem>
    <para>
      75g Kidney Beans
    </para>
  </listitem>
</itemizedlist>
```

Place the cursor behind the following line:

```
<title>Preparation</title>
```

and paste the contents of “Example 5” into your XML file.

Figure 4. HTML Output (Article With an Itemized List)

Little G's Ceylon Bang
Tux Penguin
Table of Contents
What do you need?
Ingredients
Equipment
Preparation
Abstract
Recipe for a chili.
What do you need?
Ingredients
How to prepare see the section called "Preparation"
<ul style="list-style-type: none">• 60g Habanero Chilis• 30g Cayenne Chilis• 1,5 Butch T Chilis• 75g Kidney Beans
Equipment
<ul style="list-style-type: none">• Hand blender• Pot
Preparation

Giving Step-by-Step Instructions

For writing instructions, it is recommended to do it step-by-step, as you can see in the following Example 6, “Example for Step-by-Step Instructions”. That way, instructions are easier to understand and browse for the reader.

Example 6. Example for Step-by-Step Instructions

```
<procedure>❶
  <step>❷
    <para>❸
      Rinse, than drain the kidney beans for about 10 minutes.
    </para>
  </step>
  <step>
    <para>
      Fry kidney beans with some oil.
    </para>
  </step>
  <step>
    <para>
      Puree all ingredients.
    </para>
  </step>
  <step>
    <para>
      Cook for about 5 minutes.
    </para>
  </step>
</procedure>
```

- ❶ Procedure start tag.
- ❷ If you use `<step>` elements you can create numeric listed paragraphs. Every further `<step>` you add automatically counts up.
- ❸ Within `<para>` `</para>` you can write down your text.

Figure 5. HTML Output (Article with Step-by-Step-Instructions)

Little G's Ceylon Bang
Tux Penguin
Table of Contents
What do you need?
Ingredients
Equipment
Preparation
Abstract
Recipe for a chilli.
What do you need?
Ingredients
How to prepare see the section called "Preparation"
<ul style="list-style-type: none">• 60g Habanero Chilis• 30g Cayenne Chilis• 1,5 Butch T Chilis• 75g Kidney Beans
Equipment
<ul style="list-style-type: none">• Hand blender• Pot
Preparation
<ol style="list-style-type: none">1. Rinse, then drain the kidney beans for about 10 minutes.2. Fry kidney beans with some oil.3. Puree all ingredients.4. Cook for about 5 minutes.

Integrating Remote Links and Cross References

Remote Links

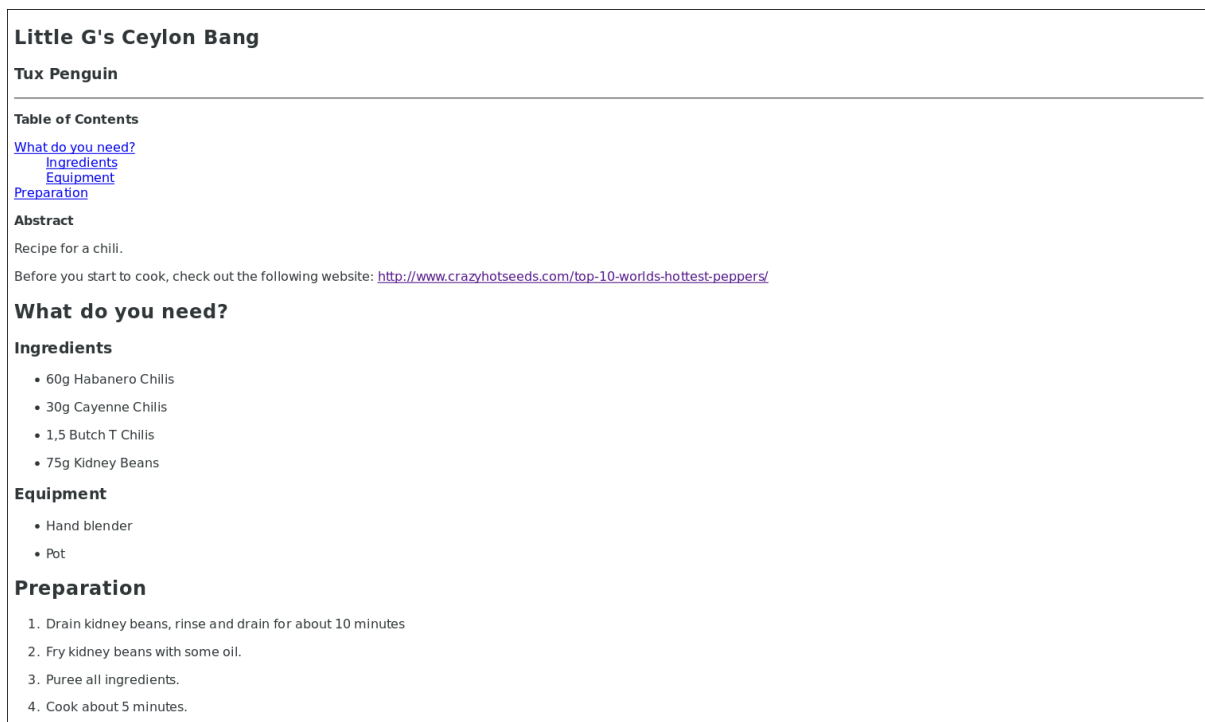
To refer to an external HTML page, integrate an internet link in your tutorial.

Example 7. Integrating Remote Link

```
<para>
  Before you start to cook:
  <ulink url="http://www.crazyhotseeds.com/top-10-worlds-hottest-peppers/" />
</para>
```

To get the output shown in Figure 6, place the cursor behind the `</para>` end tag within the `<abstract>` element and paste the content of “Example 7” into your XML file.

Figure 6. HTML Output (Article with Remote Link)



Cross References

With cross references (xref elements) you link to paragraphs or examples within your document.

Example 8. Integrating Cross References

```
<para>
  How to prepare see <xref linkend="sec.preparation"/>❶
</para>
```

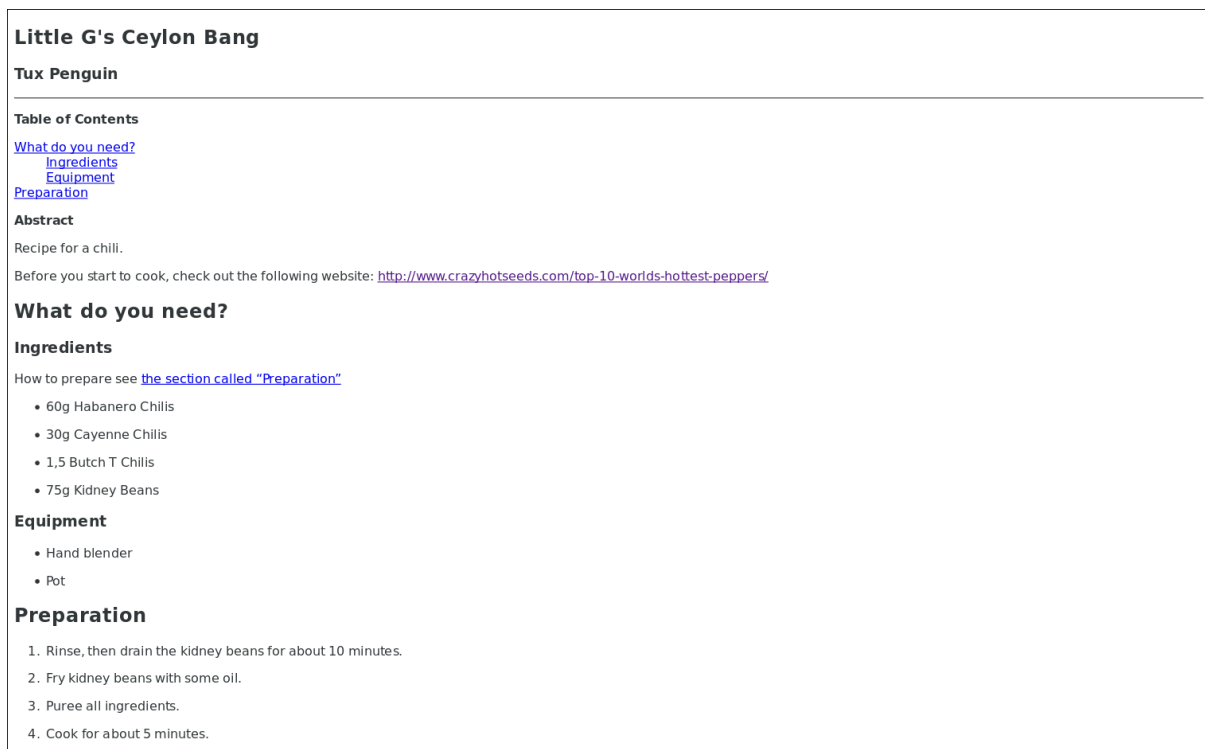
...

```
<sect1 id="sec.preparation">❷
```

- ❶ With `<xref linkend="ex.preparation"/>` you set a link to the following “Preparation” title.
- ❷ Set `id=sec.preparation` at the section, that you want to refer to. Set `id=ex.preparation` at the paragraph, you want to refer to.

To get the output shown in “Figure 7”, place the cursor below the `<title>Preparation</title>` element and paste the contents of “Example 8” into your XML file.

Figure 7. HTML Output (Article with Cross-Reference)



As you can see in the screenshot, the underlined part is a link to the section "Preparation".

Integrating Images

1. To integrate an image in your file, save an image (for example, butcht.png) in the image/png directory. That is very important because otherwise, it will not work.
2. Add an image reference into your XML file.

Example 9. Integrating Images

```
<figure>
  <title>Butch T Chili</title>❶
  <mediaobject>
    <imageobject>
      <imagedata fileref="butcht.png"/>❷
    </imageobject>
  </mediaobject>
</figure>
```

- ❶ Image title. In the output formats, the title will appear together with the image.
- ❷ The image reference.

Figure 8. HTML Output (Article with Integrated Image)

Little G's Ceylon Bang

Tux Penguin

Table of Contents

[What do you need?](#)
[Ingredients](#)
[Equipment](#)
[Preparation](#)

Abstract

Recipe for a chili.

Before you start to cook, check out the following website: <http://www.crazyhotseeds.com/top-10-worlds-hottest-peppers/>

What do you need?

Ingredients

How to prepare see [the section called "Preparation"](#)

- 60g Habanero Chilis
- 30g Cayenne Chilis
- 1,5 Butch T Chilis
- 75g Kidney Beans


Equipment

- Hand blender
- Pot

Preparation

1. Rinse, then drain the kidney beans for about 10 minutes.
2. Fry kidney beans with some oil.
3. Puree all ingredients.
4. Cook for about 5 minutes.

Figure 1. Butch T Chili



Convert Your XML Files

Convert your XML file whenever you want to, no matter if it is finished or you only want to see the temporary result. Convert as often as you like.

Important

The XML file must be valid, otherwise it will not work! Check if it is valid:

```
daps -d DC-tutorial validate
```

Convert Your XML File Into PDF

1. Open a terminal and enter the following command:

```
tux:~> daps -d DC-tutorial pdf
```

2. DAPS automatically creates a directory called build. You can find your finished PDF file there.

Convert Your XML File Into HTML

1. To build an HTML page, open a terminal and enter the following command:

```
tux:~> daps -d DC-tutorial html --single
```

2. You get a green colored link, which you can open in a browser, for example Firefox.

Further Information

If you still have questions or want to have additional information, read the DAPS man page. Open it in a terminal:

```
tux:~> daps --help
```

Or visit the following links:

- <http://www.docbook.org/tdg/en/html/docbook.html>
- <http://opensuse.github.io/daps/doc/index.html>