

# **A quick introduction to SGML, DocBook and Jade for those who just want to use it**

**Michael Hope**

**A quick introduction to SGML, DocBook and Jade for those who just want to use it**  
by Michael Hope

Quick intro to DocBook July 3rd 1998

Copyright © 1998 by Michael Hope

This document is designed for people like me who have heard about SGML and DocBook and about all the great things it can do but who don't want to wade through the language specification to understand how to write a simple document. I'm only a SGML newbie, so this is by no means complete or reliable but hopefully someone will find it usefull. It assumes only a knowledge of HTML and tags and builds from there by providing a link to the required software, a template to fill in and some pointers to more advanced texts.

# Table of Contents

<b>Preface .....</b>	<b>4</b>
<b>1. Required programs.....</b>	<b>9</b>
Introduction .....	5
DocBook 3.0 .....	5
Jade 1.1.1 .....	5
DocBook Style Sheets.....	5
<b>2. A simple template .....</b>	<b>6</b>
<b>3. Commonly used tags and their HTML equivalents .....</b>	<b>8</b>
Beginning a document - the <book> tag .....	8
Paragraphs .....	8
Sections .....	8
Links.....	8
Lists .....	9
Formatting tags.....	9
<b>4. Compiling your SGML document and some examples.....</b>	<b>10</b>
Compiling.....	10
Examples .....	10

# Preface

First up I should clear up some misconceptions that I originally had. SGML is a bit different to HTML. The SGML specification defines a language which has a structure based around the use of <tags>, just like HTML but it does not define what those tags are or what they do. You need a Document Type Definition which is an implementation of SGML to actually use it - one example is DocBook, another is the proposed successor to HTML. This however isn't enough if you want to translate the SGML into a more usable format such as HTML or RTF. As SGML is layout independent you require a stylesheet that says how to actually lay out your document on the page.

# Chapter 1. Required programs

## Introduction

To author a SGML document and compile it into RTF or HTML you need

- Document Type Definition
- SGML parser
- and Style Sheet

The ones I used are, respectively, DocBook 3.0, Jade 1.1.1 and Norman Walsh's DocBook Style sheets.

## DocBook 3.0

DocBook is supplied by the Davenport group and is available from their web site.

## Jade 1.1.1

Jade is a SGML parser written by James Clark and is available from his web site. Note that binaries are available for Win95 and NT.

## DocBook Style Sheets

Norman Walsh maintains a good set of style sheets centered around RTF and HTML for the DocBook DTD. Search for db.zip using your favourite search engine. Use the instructions in the doc directory to install and test jade and DocBook.

## Chapter 2. A simple template

SGML is significantly stricter than HTML. Most sections require other sections to be defined inside them to make them complete. For example, a chapter `<chapter>` requires a `<title>` and something inside it like a `<para>` (paragraph). And, of course, the top level section `<book>` (like `<body>` in HTML) requires a `<bookinfo>` and at least one `<chapter>` entry. The minimum required is:

```
<!DOCTYPE Book PUBLIC "-//Davenport//DTD DocBook V3.0//EN">
<book>
<title>
Your title
</title>
<bookinfo>
<bookbiblio>
<authorgroup>
<author><firstname>Michael</firstname><surname>Hope</surname></author>
</authorgroup>
<productname>
Product that this relates to
</productname>
<pubdate>July 3rd 1998</pubdate>
<copyright>
<year>1998</year>
<holder>Michael Hope</holder>
</copyright>
<abstract>
<para>
Text to appear in the abstract
</para>
</abstract>
</bookbiblio>
</bookinfo>
<toc></toc>
<preface id="PREFACE">
<title>Preface</title>
<para>
Preface with a reference <xref linkend="STYLESHEET">
</para>
</preface>
<chapter id="PROGRAMS">
<title>First chapter</title>
<sect1>
<title>First section in chapter 1</title>
<para>
This paragraph contains a list
<itemizedlist>
<listitem>
<para>Item 1</para>
</listitem>
<listitem>
<para>Item 2</para>
</listitem>
</itemizedlist>
</sect1>
</chapter>
</book>
```

```
</listitem>
</itemizedlist>
</para>
</sect1>
<sect1 id="DOCBOOK">
<title>Second section in chapter one</title>
<para>
An external HTTP url <ulink URL="http://www.ora.com/davenport/" TYPE="external">and its name</ulink>
</para>
</sect1>
</chapter>
<chapter id="TEMPLATE">
<title>The second chapter</title>
<para>
A synopsis copies the text with line breaks directly out - good for listings
<synopsis>
Blah
</synopsis>
</para>
</chapter>
</book>
```

A copy of the template is available here

## Chapter 3. Commonly used tags and their HTML equivalents

I'm no HTML or SGML wizard so this is by no means an exhaustive list. It is however enough to get you started.

### Beginning a document - the `<book>` tag

This is very similar to the `<body>` tag in HTML as it starts the proper document. Unlike HTML however you are required to add a `<bookinfo>` which in itself requires a `<bookbiblio>`, `<abstract>`. See the template for more info.

### Paragraphs

Very similar to HTML. Use `<para>` to start a paragraph. Don't forget to close it using `</para>`. Break paragraphs using `<SBR>`.

### Sections

Unlike HTML which is unstructured with the headings being a kind of afterthought, DocBook documents are rigidly structured into books, chapters and sections. This allows items like a table of contents and chapter contents to be generated quite easily and helps in dividing the SGML document into multiple HTML pages. Note that nothing should appear between sections, but can appear between a start of chapter tag `<chapter>` and the first section. The sections tags are called `<sect1>` through to `<sect4>`..perhaps further :). Each `<sect*>` must be followed by a `<title></title>` which contains the title for the section. You must do the same for a chapter.

### Links

Links are handled slightly differently in the DocBook DTD. A label is defined within a document by adding an extra field called `id` into the `sect` or `chapter` tag. For example the tag `<sect1 id="first-section">` defines a label called 'first-section'. Note that you can't use underscores in a label. To link to that label

you have a couple of choices. One is a cross reference like `<xref linkend="first-section">` will create a link reading "the section called (whatever was specified in the title) Or you can create a link where you can specify the text by using `<link linkend="first-section" type="int">the first section</link>`. to create a link to the first section with the description "the first section". Note that the type is arbitrary - I'm not even sure if it's required. External links can be made using something like `<ulink url="http://www.nowhere.com/fish/" type="ext">Go to nowhere.com</ulink>`

## Lists

I only know about `<itemizedlist>`. It's very similar to a HTML list with the `<li>` being replaced with `<listitem>`. For example the itemized list back in Chapter 1 is

```
<itemizedlist>
<listitem>
<para>Document Type Definition</para>
</listitem>
<listitem>
<para>SGML parser</para>
</listitem>
<listitem>
<para>and Style Sheet</para>
</listitem>
</itemizedlist>
```

## Formatting tags

The ones I know of are *emphasis*, *literal* and *synopsis*. `<synopsis>` will copy the text including line breaks verbatim in the courier font. The listitem example above is in synopsis style.

# Chapter 4. Compiling your SGML document and some examples

## Compiling

See the installation instructions in Norm Walsh's style sheets for installing Jade and DocBook and testing your installations. Once all three programs are installed and tested, you can compile your brand spanking new SGML file into RTF, LaTeX or HTML using Jade. For compiling into HTML I use

```
jade -t sgml -d g:\sgml\db\html\docbook.dsl file.sgml
```

For compiling into RTF format I use

```
jade -t rtf -d g:\sgml\db\print\docbook.dsl file.sgml
```

Note that for some reason the bookmarks in all of my documents dont seem to work. I dont know why.

Using some external tools you can also generate TeX, Postscript and PDF format. Unfortunately I dont have links for most of these programs as they came pre built with Debian Linux.

To turn the sgml into TeX, use

```
jade -t rtf -d g:\sgml\gb\print\docbook.dsl file.sgml
```

To actually print this file you will need the extra TeX package mentioned on the Jade pages. This allows you to produce a Postscript file using dvips:

```
latex file.tex  
dvips -o file.ps file.dvi
```

Then using Aladin Ghostscript you can turn the Postscript file into a PDF using

```
gs -dQUIET -dNOPAUSE -sDEVICE=pdfwrite -sOutputFile=file.pdf file.ps -c quit
```

I use a makefile to produce all of the above formats automatically. It looks like:

```
DB = /home/michaelh/sgml/db  
DB_PRINT = $(DB)/print/docbook.dsl  
DB_HTML = $(DB)/html/docbook.dsl  
TEX_FMT = -fmt jadetex  
  
SG=intro-to-sgml  
  
# Note: Done in this order so that the .gz's work  
all: html/book01.htm $(SG).rtf $(SG).pdf $(SG).ps.gz $(SG).tex.gz
```

```
clean:
rm -f $(SG).rtf $(SG).pdf $(SG).ps.gz $(SG).tex.gz $(SG).aux $(SG).log
rm -rf html

%.gz: %
gzip -9 $<

%.rtf: %.sgml
jade -t rtf -d $(DB_PRINT) $<

html/book01.htm: $(SG).sgml
mkdir -p html;
(cd html; jade -t sgml -d $(DB_HTML) ../$<)

%.tex: %.sgml
jade -t tex -d $(DB_PRINT) $<

# Tested with TeX, Version 3.14159 (Web2C 7.2)
%.dvi: %.tex
latex $(TEX_FMT) $<

# Tested with dvips(k) 5.78
%.ps: %.dvi
dvips -o $@ $<

# Tested with Aladdin Ghostscript 5.10 (1997-11-23)
%.pdf: %.ps
gs -dQUIET -dNOPAUSE -sDEVICE=pdfwrite -sOutputFile=$@ $< -c quit
```

And that's it!

## Examples

Strangely enough this introduction was itself written in DocBook SGML :) Some examples of SGML are:

- This document
- Documentation for some parts of GBDK.
- Digital UNIX Technical Overview at Davenports site (local copy)
- The documentation with Norman Walsh's Style sheets