Using DocBook to write GHC documentation

The GHC Team

January 2000

Table of Contents

1. Getting the DocBook tools

See the installation guide.

2. Document layout

The GHC documentation is written using DocBook XML V4.2, so the first few lines should look like this:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
   "http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd">
```

The encoding can of course be chosen according to taste.

This guide is *not* meant to teach you how to write DocBook; read the DocBook book [http://www.docbook.org/] for that. It is more of a reference than a tutorial, so see the DocBook home page [http://www.oasis-open.org/docbook/] for other links.

However, by popular demand, here are some useful points:

- Remember to use <para> inside <listitem>s.

The rest of this section outlines the use of several tags which may not be obvious (DocBook is rather scholastic in style: it has tags for many things from C function prototypes to keyboard bindings; at the same time it has many omissions and oddities). The current scheme has many infelicities, partly because it was dreamt up in a hurry while the author was learning DocBook and converting the documentation thereto, and partly because DocBook is rather C-centric.

Comments
: Comments in XML look like this: <!--This is a comment-->.

<command>
: Used for commands typed into interactive sessions (e.g. **cp foo bar** and the names of programs such as **gmake**.

`<constant>`	Used for system constants such as `U_MAXINT` and `Makefile` variables like `SRC_FILES` (because they are usually constant for a given run of **make**, and hence have a constant feel to them).
`<email>`	For email addresses. This is a tag that's easy to overlook if you don't know it's there.
`<filename>`	Used for paths, filenames, file extensions.
`<function>`	Used for functions and constructors.
`<indexterm>`	The normal way to mark up an index term is `<indexterm><primary>term</primary></indexterm>`.
`<keycap>`, `<keycombo>`	Some more tags you may miss. Used for combinations such as **Control**-**D**.
`<literal>`	Used for everything that should appear in typewriter font that has no other obvious tag: types, monads, small snippets of program text that are formatted inline, and the like.
`<option>`	Used for compiler options and similar.
`<programlisting>`	For displayed program listings (including shell scripts).
`<screen>`	For displayed screen dumps, such as portions of shell interaction. It's easy to tell the difference between these and shell scripts: the latter lack a shell prompt.
`<varname>`	Used for variables, but not type variables.

3. Tables

Tables are quite complicated to write in DocBook XML (as in HTML, there are lots of fiddly tags), so here's an example you can cannibalise. In the spirit of the LaTeX short introduction I don't repeat all the markup verbatim; you have to look at the source for that.

Here's	a sample	table
With differently	aligned	cells

There's not much else to it. Entries can span both extra rows and extra columns; just be careful when using block markup (such as `<para>`s) within an `<entry>` that there is no space between the open and close `<entry>` tags and the adjacent text, as otherwise you will suffer from Pernicious Mixed Content [http://www.docbook.org/tdg/html/entry.html] (the parser will think you're using inline markup).