Quick SGML Example

Matt Welsh, mdw@cs.cornell.edu v1.0, 28 March 1994

This document is a brief example using the Linuxdoc-SGML DTD.

Contents

1 Introduction

This is an SGML example file using the Linuxdoc-SGML DTD. You can format it using the command

```
% sgml2txt example.sgml
```

this will produce plain ASCII. You can also produce LaTeX, and HTML and GNU info.

2 The source

Looking at the source for this file will be instructive to show you how to use many of the Linuxdoc-SGML constructs. You should also read the *Linuxdoc-SGML User's Guide*, in the file `guide.sgml`.

The source looks and feels like LaTeX, as you can see. Paragraphs are separated by blank lines, macros are enclosed in angle brackets. It's quite simple.

3 Some examples

Here are examples of things that you'll be using in the source. First, let's decend into a subsection:

3.1 This is a subsection

As the header says. Note that you need to use the `p` command to start the body of the section, after the `sect1` command. Here's a subsubsection:

3.1.1 This is a subsubsection

Right. 5 levels of sections are available. Use the commands `sect`, `sect1`, `sect2`, `sect3`, and `sect4` to get them. This document uses the `article` document style, which is appropriate for HOWTOs and other docs; the `report` style (which includes the `chapt` sectioning command) should be used for the LDP docs.

3.2 Example text

All right, so you're typing along, and you want to show some example code, or example I/O with a program, whatever. Use the `code` or `verb` "environments" for this, wrapped in a `tscreen` environment, as so:

```
This is an example verb environment.
```

As well as:

```
This is an example code environment.
```

The `tscreen` environment just sets the font to small type and indents it nicely. It's not required for using `verb` or `code`, but I suggest that you do.

The *Linuxdoc-SGML User's Guide* explains what special characters you can and can't use in the `verb` and `code` environments.

3.3 Cross references

What about cross-references? This section has been marked with the `label` command; using `ref` will provide a cross reference, as in "See Section 3.3 ()" for more.

Right now cross-references don't work in the `groff` translation for plain ASCII. They do work when generating LaTeX or HTML output.

3.4 Using fonts

You want fonts, we got fonts. Of course these won't show up in the plain ASCII text, but they all map into the various output formats: **boldface**, *emphasis*, sans serif, *slanted*, `typewriter`, and *italics*.

3.5 Lists

Lists are easy as well. Just use the `itemize` element with the `item` commands, seen here:

- This is a list.
- Nothing exciting about that.

  - Multiple levels are supported as well.
  - Again, that's no surprise.

1. Enumerated lists using `enum` also work.
2. Description environments using `descrip` along with `tag` are also available, as seen here.

First item.
Here's one item.

Second item.
Here's another.

Third item.
Can we go for three?

- A final item to top it all off.

This is just about everything that you need to get started with writing SGML docs using the `linuxdoc-sgml` DTD. Please let me know if you think something should be changed or added to this document.