Hypertext Style Guide

timbl@info.cern.ch

April 14, 1993

Contents

1 Style Guide

This guide is designed to help you create a hypertext database effectively communicates your knowledge to the reader. It has been prepared in the light of comments by readers, and many demands by providers on online documentation. Some of the points made may be influenced by personal preference, and some may be common sense, but a collection of points has been demanded, and so here it is.

The guide is designed to be read sequentially, but feel free to depart from this. The sections are as follows:

- Introduction
- Overall structure of your work
- Within each document
- Test your document
- Background reading

1.1 This document is under construction!!!!!!!!

Lost of peices are missing – and some of them haven't even been thought of! Suggestions are strongly invited, if you think of anything mail it to timbl@info.cern.ch, mentioning the Style Guide for Online Hypertext.

2 Introduction

You are going to write (or generate) some online hypertext. Because hypertext is potentially unconstrained you are a little daunted. Do not be. You can write a document as simple as you like. In many ways, the simpler the better.

You will be writing a number of separate files. These files will be linked to each other, and to external documents, to make your final work.

You may think of your work as a "document", and if it were on paper, then you would call it that. In the online case though, we tend to refer to each individual file as a document. A document may correspond, in the book analogy, to a section or a subsection, or even a footnote. In this guide, we'll refer to the whole collection as a work.

The document is the unit by which information is picked up. At any one time, a document is completely loaded into the reader's computer. It is also normally the amount you edit at any one time, though with a good editor you will probably have a number of documents open at a time.

The section on structure discusses how you organize your material into documents. Another section discusses how to organise your material within a document.

(Up to overview , on to structure)

3 Structure

If you have in mind a body of information to put across to your reader, you probably have a mental organisation for it. Normally this is a sort of hierarchical tree, like the chapters of a book if you were to write a book.

Keep this structure. It helps readers to have a tree structure as a basis for the book: it gives them a feeling of knowing where they are. You can also us this structure for oganising your files in directories.

You should also bear in mind:

- The reader's preconceived structure
- The idea of overlapping trees
- How big to make each document

(Up to overview , back to Introduction, on to: writing each document)

4 The reader's structure .

Remember always the audience for whom you are writing. If they are novices in the subject, it will normally help if you are firm about the structure of your work, so that they can learn the structure of the knowledge itself. For example, if you feel that the subject falls into three distinct areas, then that is an importnat thing to teach.

If, however, your readers will already have some knowledge in the subject, then they will already have formed their own structure for it. In this case they will conciously or subconsiouly know where they expect to find things. If your structure is different from theirs, enforcing it too strongly will confuse them and put them off.

You may in this case have resist a strong tendency to put across your own structure strongly and to the detriment of all others. There are two solutions.

If you have a single well-defined audience in mind, who will share a similar world view, then try to write excatly for that world view rather than yours.

If you are simultaneously writing for more than one group, then you must provide for both.

When you make a reference, qualify it with a clue to allow soime people to skip it. For example, "If you really want to know how it works inside, see the Internals guide", or "A step-by-step introduction is in the tutorial".

Provide links for both reader's views. Your work will be more connected than a simple tree, but with proper qualifiaction, noone should get lost.

Provide two sepate tree "roots". For example, you can write a step-by-step tutorial and a functionaly direct reference tree for the same data. Both will at the lowest level have the same data, but while the first will deal with the simple things first, the second may be functionnaly grouped. This is just like having several indexes to a book. The tutirial might also include information which the reference work does not.

(Up to overview , back to Introduction , on to: writing each document)

5 Overlapping Trees

Here is an example of a work (describing some programming functions, say) with two separate structures:

```
        Tutorial                          Reference
           |                                  |
    Let's do it together              -----------------
  from simple to difficult            |                |
           |                   by Functional     Alphabetical
           |                       group            by name
    Task oriented examples            |                |
           |                          -----------------
           |                                  |
    Examples of use of              Syntax definition for
    specific functions   <-------->   specific functions
```

The novice user starts at the top left, and works his way down. Where he needs specific details, he will get down to the examples and from them a link to the underlying definitive desctiptions of each. As far as he is concerned, he is reading a tree-strucured work. In fact, he is reading the same information as the expert who, coming in to check on one particular function, then looks up an example of its use. (Up to structure , back to user's structure , on to: document size)

6 How big to make each document

The most important point here is that a document should put across a well-defined concept. It is not generally worth splitting one idea arbitrarily into two bits in order to make the bits smaller. Nor is it a good idea to put together ideas which area really separate just to make a bigger document.

A document can be as small as a footnote .

There are two upper limits on a document's size. One is that long document swill take longer to transfer, and so a reader will not be able to simply jump to it and back as fast as he or she can think. This depends a lot on the link speed of course.

The other limit is the difficulty for a reader to scroll through large documents. Readers with character based terminals don't general read more than a few screens. They often only absorb what is on the first screen, as if that is not interesting they won't be bothered to scroll down.

Readers with graphic interfaces generally scroll through long documents with a scroll bar. When the scroll bar is moved a small amount, the document should move a sufficiently small amount so that some of the original window-full is still left in the window. This allows the reader to scan the document. If the document is any bigger, then it is basically unreadable, in that any movement of the scroll bar will loses the place and leaves the reader disoriented.

(In fact, one can normally scroll around explicitly page by page, but this is gives the same feeling as the terminal interface.)

A rough guide, then, for the size of a document is:

- For online help, menus giving access to other things: small enough to fit on 24 lines. Check this by using a terminal browser.
- For textual documents, of the order of half a letter-sized (A4) page to 5 pages.
- (Up to structure, back to overlapping trees, on to: within each document)

7 Within each document

This section of the style guide deals with the layout of text within a "document", the unit of retrieval of information on the web.

To be completed.

You should try to:

- Make links into context .
- Use context-free document titles
- Format device-independantly
- Write for the printed work too

(up to overview , back to structure , on to testing)

8 Linking to context

A major difference between writing part of a serial text, and an online document, is that your readers may have arrived from anywhere. Even though you have only made links to it from one place, others may want to refer to that particular point, and so make a link to that particular part of your work from their own. So you can't rely on your reader

Of course if you are writing a tutorial, it will be important to keep the flow from one document to the next in the order you intended for its primary audience. You may not wish to cater specially for those who jump in out of the blue, but it is wise to leave them with enough clues so as not to be hopelessly lost. Some ways of doing this are:

- Watch that your text and vocabulary stands by itself. Starting a document with "The next thing we we consider is..." or "The only solution to this problem is..." will certainly confuse.

- Sometimes the opening words refer to the context, and can be linked to background information. For example, in the WWW project documentation, the first occurence of the acronym WWW is often linked back to the central project document.
- The navigation hints at the top or bottom of the document can give explicit pointers. Examples are at the bottom of this document.

It can also be useful to imagine as you are writing that you yourself may wish to reuse the document.

(Part of style guide for online hypertext. Up to Writing each document, on to ...)

9 Device Independence

The hypertext you write is stored in HTML language, which does not contain information about the fonts and paragraph shapes and spacing which should be used for displaying the document.

This gives great advantages in that your document will be rendered successfully on whatever platform it is viewed, including a plain text terminal.

You should be aware that different clients do use different spacing and fonts. You should be careful to use the structuring elements such as headers and lists in the way in which they were intended. If you don't like the rendering on your particular client, don't try to fix it by using inappropriate elements, or trying for example to force extra spacing with empty elements. This may well end up being interpreted differently by other clients and looking very strange. You can in many cases configure the client displays each element.

For example:

- Always use heading levels in order, with one heading level 1 at the top of the document, and if necessary several level 2 headings, and then if necessary several level 3 headings under each level 2 heading. If you don't like the way heading level 2 is formatted, fix it on your client, don't just skip to heading level 3.
- Don't put extra spaces or blank lines into your text to pad it out, except in preformatted (PRE) sections.
- Don't refer in your text to facets of particular browesrs. Asking someone to "click here" won't make sense without a mouse, just as asking someone to "select a link by number" will betray the fact that you were using the line mode browser. Just leave a link. The instructions get boring as the user will normally know how to select a link.

See also: testing your document .

Following these guidelines you may find that the end result does not appear on your screen exactly as you would like, but your readers will probably be happier.

(Part of the Style Guide for Online Hypertext . Up to within each document , back to , on to printable hypertext)

10 Printable hypertext

In an ideal world, paper might not be necessary. In a next to ideal world, one would have enough time to write a hypertext version of a document and also a completely reauthor a paper version. In the real world, you wilkl probably want to generate any printed documents and online documents from the same file.

Suppose the HTML files will be the master, and you will generate the printable from this, by translation into TeX, etc.

If you might one day want to do this, try to avoid references in the text to online aspects. "See the section on device independence" is better than "For more on device independence, click here.". In fact we are talking about a form of device independence.

Unfortunately the recommended practices of signing each document and giving navigational links tend to mess up the printable copy, though one can of course develop ways of stripping them out if they follow a common format.

(Up to: within each document; back to device independece, on to ...)

11 Test your document

In a way your hypertext is like a book, which you should have proof-read. In a way, it is like a program which you should have tested. At least get someone from the group for which you wrote the document to read it and give you some feedback. Other ideas are:

- Read the document several different client programs, to ensure that you have formatted it in a device independent way.

- Monitor the readership of your document. You can do this by analysing the server log files . You may find that some parts are not being read, perhaps because people are looking in the wrong place for them. You may see that people often follow a path and backtrack. If you can guess what they were looking for, you can make the clues around the link more helpful. (Remember to keep log information confidential until you have removed user information from it.)

- Make it clear whether your will accept criticism or suggestions from your readers, and how they should send it.

- Ask people to solve problems using the document, and report on their success. If they fail, find out what they were looking for, whether it was in the document at all,

11.1 How much testing?

Testing takes time. The decision of how much testing you do is based on the quality of the document you wish to provide. You are balancing your reader's time and effort against yours. If your document is "selling" an idea, or if you are selling the document or providing a service, you will want to make it as easy as possible for the reader. If many people will read your work, a little of your time will save a lot of theirs.

If however you are documenting some obscure part of a system in which no one other than yourself is likely to be interested, or if you feel that your readers are lucky to have anything available at all, there is no point wasting time testing it. In the event of someone needing the information, they might have to go to some extra trouble to follow several links to find what they want, and then to understand what you have written. This may be the most efficient way of working. I emphasize this because there is very much information which is for a fleeting moment in people's minds, or is hastily scribbled down on some file, and which may be important to posterity. It is better for this information to be available even in unpolished form than for it to be hidden out of embarrassment for its form. Before electronic technology, the effort of publishing was such that this information was never seen, and it was a waste, and and considered an insult to one's readers, to publish something which was not of high quality. Nowadays, there is "publishing" at all levels, and both high quality and hasty documents have their value. It is important, though, to make it clear what the quality of a document is when making a reference to it, to avoid disappointment.

Monitoring the server log files will tell you which documents are really being read. You can use your time most efficiently to improve the quality of those. Of course, analysing the server log files also takes time!

(Part of the Style Guide for Online Hypertext. Back to ..., On to ...)