

Introduction



Campbell Laird/Stock Illustration Source

Single Source

Part 2

BY PHILIP BUTLAND

Editor's note: Part 1 of this article, which explored the advantages of single sourcing and discussed the differences between manuals and help, appeared in the February 2001 issue of Intercom. Part 2 looks at obstacles to single sourcing and how to overcome them.

In Part 1 of this article, I said that single source is much more than just using one document to produce manuals and help. I described how, with some investment into company- or department-wide software, single source principles can minimize duplication of work across departments.

Most of the work in single sourcing

comes at the beginning of the project. Let's say, for instance, that you're planning to use a single source to produce both a printed manual and online help. You need to figure out which parts of the single source file will go into each. (It's very rare that all the copy from a manual will go into the online help and vice versa.)

As you plan the contents of your manual and help, you must think carefully about structure. This is a big effort, but it will pay off in the long run, especially if you use good single source software. If you do, you only need to obey a few rules, and everything should work fine.

About the Examples

I've illustrated the examples below using techniques from the software package *SmartDoc*, produced by t3-medien in Germany. *SmartDoc* is software that inte-

grates WebWorks *Publisher* facilities with *FrameMaker* menus. *SmartDoc* is promoted as being about more than software: 50 percent of the package concerns a method of writing single source documentation, for which t3-medien provides comprehensive training.

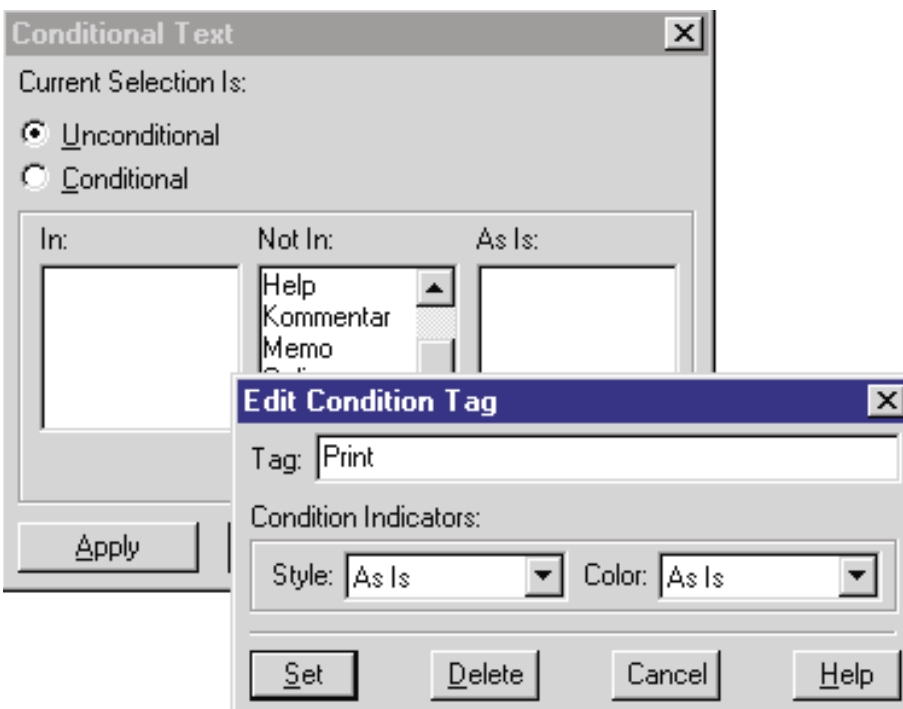
In some instances, I've assumed you're familiar with these software packages. By using these examples, I am not endorsing *SmartDoc*, *Publisher*, or *FrameMaker*. It's my hope that the features of *SmartDoc* will be similar enough to your software that you can easily reproduce my examples on your system.

Using Conditional Text

There are times when you'll want to use *conditional text*—text that appears in either the printed or online documents. How do you specify such text?

The simplest way of differentiating between manual text and help text is to use the *FrameMaker* conditional text function (see Figure 1). First, you need to create two conditions: *print* and *online*. You create a condition by selecting **Special—Conditional Text**, then clicking the Edit Condition Tag button. Enter “print” or “online” in the Tag Edit field, and *FrameMaker* creates a new condition.

Figure 1. Setting Conditional Text



You can now mark up conditional text by selecting it in the *FrameMaker* source file, then selecting **Special—Conditional Text**. Use the In, Not In, and As Is fields to specify the conditions of the selected text.

Pressing the conditional text Show/Hide button allows you to specify the conditions applying to a document. Single source software can be automated so that help files do not contain text marked as “print” and printed files do not contain text marked as “online.” Text that is not marked up appears in all documents. For more information about conditional text, please consult your *FrameMaker* documentation.

Contents of Book Files

When you are creating help and manuals from the same source with *SmartDoc*, you can create two separate book files—the master files for printed or online documents. These files can contain exactly the same body chapters (although they do not have to; for example, if you want to provide detailed information in help, you may omit some chapters from the manual book file). They should, however, contain different first chapters. Because the opening page of a help file and the

Information on Single Source Software

For more information on single sourcing, refer to the following sources.

www.quadralay.com has evaluation copies of WebWorks *Publisher*.

www.wwpusers.com is a user group for WebWorks *Publisher* users.

www.scriptorium.com has a number of useful presentation documents.

www.t3medien.de is the Web site for t3-medien, developer of *SmartDoc*.

front matter of a manual are different, there is little overlap in the information they contain. Each first chapter can include medium-specific information, such as the chapter structure of a manual or the page layout of the help. Also, the opening chapter is a useful place to store markup information, such as conditional text settings. This markup information can determine how the manual and help file will appear, and will therefore be different for different media.

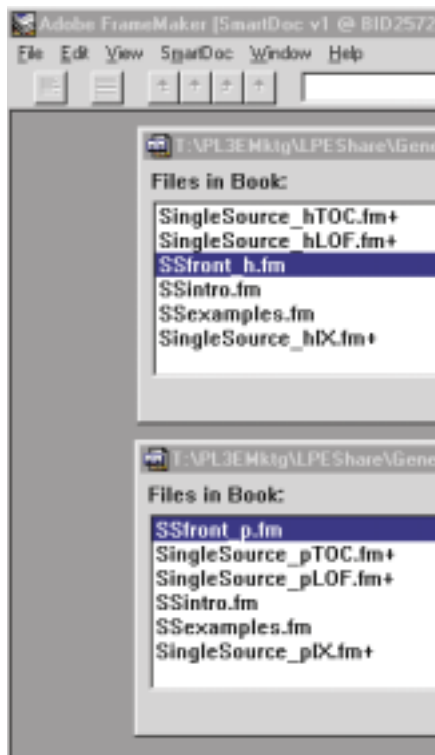
The files generated by *FrameMaker*—the table of contents, the list of tables/figures, the index, and so on—will also be different. This need not bother you, since these files are generated automatically from the book files. Figure 2 shows book files for help and print. The chapter files *SSintro.fm* and *SSexamples.fm* are common. The generated files (which end with a +) are automatically assigned different names and are in a slightly different order. Only the front matter files *SSfront_h.fm* and *SSfront_p.fm* are completely different.

Sarah O’Keefe, a consultant who has worked on a variety of single sourcing projects, has suggested an alternative solution to the problem of formatting manuals and help: Use a single book file, and manage the files and settings with WebWorks *Publisher*.

Using Links

If you choose neutral language (discussed in Part 1 of this article), the main difference between manuals and help will be in the formatting. The words on the page will be the same, with the exception of links. Links in a manual will generally

Figure 2. Book Files for Help and Print



refer to page numbers, which have no meaning in online help. Help, on the other hand, can use hyperlinks, which are unavailable in printed manuals, although they can be used to some extent in PDF files. Use the *FrameMaker* **Special—Cross-Reference** command to set up links. Single source software can process these links so that they appear differently in manuals and in help.

Essential Link

The most common link in help is a so-called Essential Link: You click on the text, and the help jumps to the referenced page. In the manual, you see a page reference; PDFs contain a combination of the two. For example, your manual may contain the text “For more information, see ‘Specifications’ on page 26.” This instruction obviously makes no sense in a help file, where you would probably see “For more information, see the specifications,” with “specifications” marked up as a hyperlink. In the PDF file, you see the same text as in the manual, but you can also click on this link to jump to the referenced page.

You can use the *FrameMaker* cross-reference feature to specify that you only

need enter a single link in the source file. When the manual and help file are generated, the links will automatically appear in a form appropriate for the medium.

Optional Links

Optional links appear in help systems and provide additional information that may not be available in a printed manual. For example, you might have text that reads, “You should now connect a fiber.” In a help file, you can create a hyperlink from “fiber” to more detailed definition. Clicking on “fiber” in the help file will bring up a screen giving extra information. You might want the PDF to contain a link to a glossary, but such information in a manual might confuse users and disturb the flow of your text. Again, you can set up your *FrameMaker* cross-reference definitions so that the appearance of your text is appropriate for the medium.

Related Topics

Some links are not part of the text at all. They appear as a list of references at the bottom of a page, or are accessible by a button at the top of a page. These links reference pages that give related information. Known as Related Topics Links, they are usually available in help files only.

There are several types of Related Topics Links. The ones I use most often are

buttons at the top of a help screen. If users click on these buttons, they see the titles for a number of related topics. (See Figure 3.)

A selection of buttons at the top of a help screen lets you know in advance what sort of information you’re going to see. I use the following types of buttons:

- *Basics*: A basic introduction, suitable for a beginner.
- *Details*: Detailed information aimed at more experienced users.
- *How to*: A step-by-step procedure on how to do something.
- *Examples*: An example of how something works.

You can set up these buttons when you write the text for this segment. If there are no links of a particular type, that button will not be shown.

Setting Up Context IDs

You use context IDs if you want to produce context-sensitive help (that is, help that appears when you press a help button or *<F1>*). Each context-sensitive screen has at least one ID, which is stored in a resource file. Resource files usually have the suffix *.H* or *.HM*.

A context ID can be a decimal number, a hexadecimal number, or a string. If

Figure 3. Using Related Links



you're not sure what your context IDs should be, have a word with your software engineers. The way you set up context IDs depends on the single source software that you're using. With *SmartDoc* software, write the string in a paragraph of format `z_context_id`. Once you have associated a context ID with a particular screen, recompile the software with which this help is running. If everything is working properly, a help button or `<f1>` will bring up the referenced screen.

Searching for Text

The search facilities in manuals and help are different, and are used in different ways. Single source software allows you to use the indexing facilities that you find in a normal help file. For example, your CHM help file may contain the following index and search options:

- *Contents*: An ordered list of screens in the help file, essentially the same as the TOC file generated by *FrameMaker*. (See Figure 4.)
- *Index*: An ordered list of index words, essentially the same as the IX file generated by *FrameMaker*.

- *Search*: Searches the help file for all occurrences of a specified word or phrase.
- *Favorites*: Remembers a page that you can return to at a later stage.

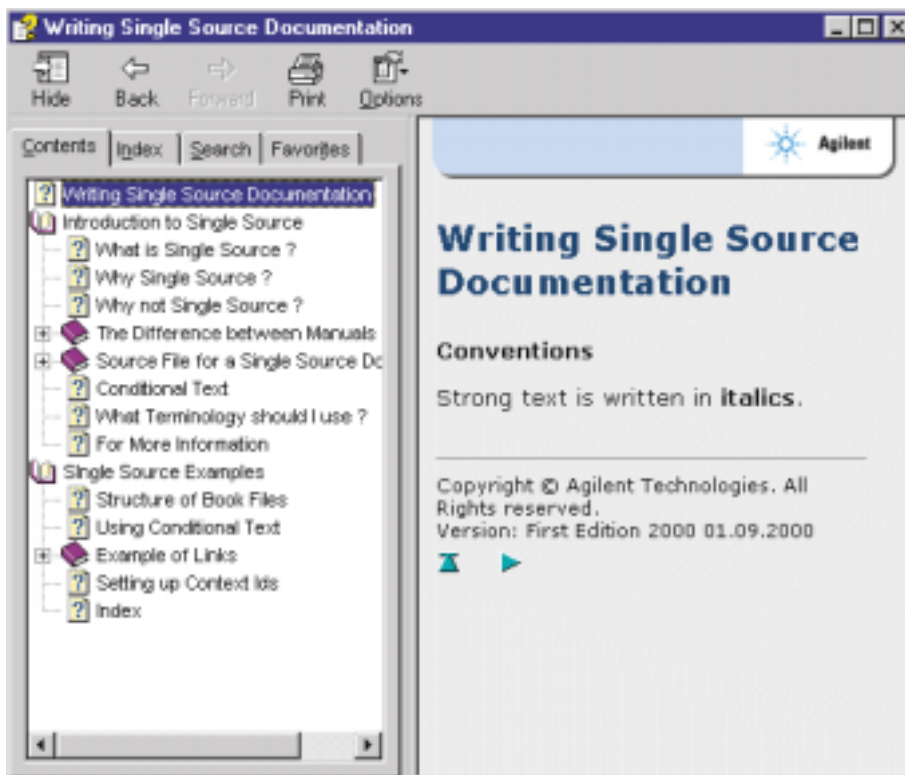
All of these options can be automatically generated by single source software. The first two are generated by selecting a table of contents and an index in the book file for your help.

Where Next with Single Sourcing?

So far, I have discussed how single sourcing can be used with relatively small impact on the way we normally work. If single source is going to be a major component of our procedure for producing documents, it may affect the way we work more profoundly. Most important, it offers opportunities for building our own databases.

I recently started a project with the marketing communication department at Agilent Technologies, my employer. The project aims to build a database of common resources, with the purpose of regulating what we put on the Web. I am also working on a similar project with the technical writing community across Agilent. Both projects were initially independent

Figure 4. CHM Contents Page



Single Source Tools

If you are considering single sourcing, you may want to look at some of the following software. Inexpensive single sourcing tools include the following:

- *Doc-To-Help* (WexTech Systems)
- *ForeHelp* (Forefront, Inc.)
- *FrameMaker* (Adobe Systems, Inc.)
- *RoboHELP Office* (Blue Sky Software)
- *SmartDoc* (t3-medien)
- *WebWorks Publisher* (Quadralay Corporation)

Content management systems:

- *Canterbury* (Chrystal Software)
- *Content@* (XyEnterprise Enterprise Solutions, Inc.)
- *Directive* (HyNet Technologies)

XML-based tools:

- *Astoria* (Chrystal Software)
- *Epic* (ArborText, Inc.)
- *POET* (POET Software Corporation)

Enterprise technology:

- *Enigma* (Enigma, Inc.)
- *XyEnterprise* (XyEnterprise Enterprise Solutions, Inc.)

If your budget is limited, then focus on the first group. *Canterbury*, for example, starts at around \$40,000. Other tools listed in this group have similar pricing.

of work that I was doing with single sourcing, but they do not have to be.

As databases are built, some thought must be given to what content goes into them and how that content is utilized. Single sourcing can be used not only locally, but also as a blueprint for our wider work. **1**

Philip Butland is a learning products engineer at Agilent Technologies in Germany. He is a member of Agilent's Learning Products Council and is currently working on the design team for Agilent's manuals and help files. For general information or feedback on this article, please e-mail him at Philip_Butland@Agilent.com.