

# Single-Sourcing with FrameMaker and WebWorks Publisher



**Scriptorium Publishing Services, Inc.**  
P.O. Box 12761, Research Triangle Park, NC 27709  
919-481-2701 or [sales@scriptorium.com](mailto:sales@scriptorium.com)  
<http://www.scriptorium.com>

## Introduction

*Single sourcing* refers to a writing process in which you create different deliverables from a single set of files. With today's technical publishing tools, it's become easier to create multiple outputs—including hard copy and online help—from one set of files.

Single sourcing takes quite a bit of planning. However, if you do it right, you can create high-quality deliverables *and* save money and time. If you are the lone technical writer in your company or if your department is understaffed, single sourcing could be the only approach that makes it possible to deliver your books and help on time and within your budget.

This discussion assumes that the content in your printed books and online help is identical or has a significant amount of overlapping information. If the content that you deliver in books is different from the content that you deliver online, single sourcing doesn't make any sense. Instead, you'll develop the online help and the books separately, using whatever tools make the most sense for each deliverable.

The FrameMaker/WebWorks Publisher combination offers a cost-effective, powerful way of delivering print, PDF, HTML, WinHelp, and other online help formats from a single set of FrameMaker source files. This approach requires highly structured FrameMaker templates and a disciplined approach to writing, which make it possible to deliver high-quality single-sourced deliverables.

## The traditional workflow

Without single sourcing, a documentation department would typically use one of two approaches to delivering printed and online materials:

- Developing deliverables simultaneously (parallel development).
- Developing first one deliverable, then the other (serial development).

## **Parallel development**

Parallel development means that the books and the online help are created, in separate files, at the same time. Often, this means that one technical writer creates the book and another creates the online help.

This approach has a few advantages:

- Both deliverables are ready at the same time.
- Writers can specialize in print or online work and optimize the information for each deliverable.

But there are also a number of disadvantages:

- This approach is very labor-intensive and maintenance is problematic because you're creating the same information twice. This duplication of effort can also lead to differences in terminology between the printed and online versions, which confuses the readers.
- Often, the print tool and the online help tool do not work well together. If you need to transfer information from one tool to the other, formatting is lost.

Parallel development makes sense when the information in the print and the online help does not overlap. But when there is overlap, parallel development becomes time consuming and inefficient.

## **Serial development**

Serial development could be considered single sourcing. You write the information once, and then convert it from one format to another. However, serial development typically involves copying the information from one format (for example, the print development tool) to another format (the help authoring tool) and then reworking the information to add formatting.

Serial development has some advantages:

- Because the information is written once, information is consistent across all deliverables.
- The second deliverable is not created until the first deliverable is completed, so the information is finalized.
- Maintenance is simplified because you only convert once per release and do not have to maintain two sets of documentation.
- One writer can first create the print, then the help, so it's less expensive than having two writers working in parallel.

But there are some serious disadvantages:

- Serial development means that one deliverable will lag significantly behind the other. For example, the printed version might be ready three or four weeks before the online help. This leads to scheduling problems before a release because you have to build in several weeks for the conversion and reformatting process.
- Often, the print tool and the online help tool do not work well together, so formatting is lost when you transfer information from one tool to the other. The cleanup that's done in the second version must be repeated for each release (unless you keep the files and only put in the changes, in which case you've switched to a parallel development process).

Many companies use serial or parallel development. However, single sourcing provides them with another option for creating help and other deliverables. In a single-sourcing workflow, you create a single set of files that contains all the information for both the printed and the online version, and use features such as *conditional text*—text that you tell your text development tool to show or hide based on tags you apply to the text—to control which information appears in which deliverable.

Figure 1 shows a single-sourcing workflow for FrameMaker.

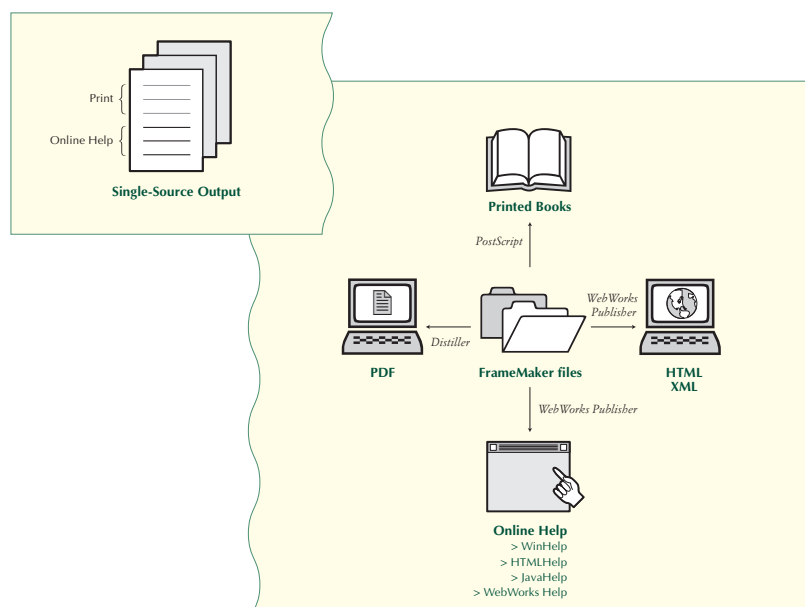


Figure 1: A FrameMaker-based single-sourcing workflow

## Evaluating whether single sourcing is right for a project

Single sourcing is not appropriate for every project. If the content of the various deliverables does not overlap, there's little reason to use single-sourcing techniques. For example, if the content of your online help has little in common with your user guide, there's no sense in trying to create the help and the printed user guide from one file set.

## Benefits of single sourcing

If your project has significant content overlap from one output medium to another and if you have budget and time constraints, single sourcing can produce benefits for your documentation effort. Benefits of single sourcing include the following:

- Eliminating maintenance on multiple file sets.
- Saving money.
- Presenting information customized for each delivery medium.

All of these benefits make it easy to build a business case for single sourcing.

### Eliminating maintenance on multiple file sets

Because print and help are generated from the same source files, you no longer have to maintain two (or more!) sets of content in a parallel development process.

Maintaining the same information in two locations is obviously time consuming (you have to make every update twice), but there's another, less apparent cost. Maintaining multiple information sets increases your error rate because you double the chance of making mistakes.

If you have to make 100 changes to your content, you would make 100 changes in a single-sourcing environment. But if you're maintaining two sets of source files, you have to make 200 changes. Or, more accurately, you have to make 100 changes two times. This is extremely tedious for the writer and can lead to additional errors in your content.

### Saving money

At a minimum, single sourcing eliminates the problem of making updates in two places. This alone can provide significant savings.

But you can also eliminate the costs associated with maintaining multiple file sets for different output formats—and this can make the other savings look puny—especially if your content is later translated into other languages.

You also save money because of increased efficiency and cleaner files.

### **Presenting information customized for each delivery medium**

In addition to saving money and eliminating the need to maintain multiple file sets, single sourcing lets you to develop a template for each delivery medium that takes full advantage of that medium. For example, a glossary might be delivered as a series of pop-up topics in online help, but it can be the standard alphabetical list in a book. Reference information can be hyperlinked and made searchable online. Detailed conceptual drawings are probably best left in the printed, high-resolution book. But in online help, hyperlinked flow charts are possible.

### **Making the business case for single sourcing**

Consider the case of a typical software company. The company has approximately 6,000 pages of documentation in a total of 10 books. The material is translated into eight languages.

The writers develop books in FrameMaker and then save the FrameMaker files to rich text format (RTF), which preserves fonts, italics, and other formatting. The writers import the RTF files into RoboHelp, and clean up formatting problems to create WinHelp. Typical formatting tasks include reimporting graphics (which are lost during conversion), recreating hyperlinks and cross-references (lost during conversion), breaking material into topics, creating pop-up topics, and correcting numbering problems. On average, the cleanup process for a 200–300 page book takes about three weeks (120 hours). At a conservative \$30 per hour cost, that works out to \$36,000 in conversion costs for 10 books. Therefore, the conversion costs for the English source and all eight languages means a total of \$324,000 (9 languages x \$36,000)—and that doesn't include translation costs.

The company now introduces a single-sourcing process. The conversion to online help is handled by converting the source files to online formats with an automated mapping tool. The conversion process takes just one hour per book! The company incurs approximately \$13,000 in one-time costs for training, template development, software licensing, and consulting. Even

with a much higher conversion rate of \$120 per hour (versus the \$30 for the RoboHelp conversion), the recurring conversion cost is only \$10,800 (1 hour x \$120 per hour x 10 books x 9 languages). Translation costs aren't included in this total, either.

The old process costs about \$324,000 per release. The new process costs about \$23,800 for the first release—and that includes the \$13,000 in one-time setup costs.

In short, the cost savings are enormous. If this company produces two releases per year (one every six months), the single-sourcing solution will save more than half a million dollars in the first year!

## **Objections to single sourcing**

Single-sourcing opponents believe that it's impossible to write content that works well both in printed and online documents—they say that the presentation requirements for print and online are too different for a common source file. It's certainly true that some single-sourcing efforts have produced mediocre results. For example, you've probably seen online help that read more like a book—windows full of conceptual information that a user really doesn't want when trying to figure out how to perform one task, and text that didn't take advantage of online hypertext linking.

In our experience, it's quite possible to write content that works in several output formats. Extensive planning and using the right tools can give you complete control over the appearance of both printed and online materials.

Although it's probably true that “hand-crafting” printed and online materials separately will result in higher-quality results, a solid single-sourcing process can produce very good results at a fraction of the cost. This is the 90 percent point. If you can achieve 90 percent of the “hand-crafted” quality at a fraction of the price, you must decide whether the cost you incur to achieve that last 10 percent is worth the effort.

## **Planning for single sourcing**

Planning should be an important part of any writing project, but for a single-sourcing project, it's required. Because the book components will later become online help components, the writer (or template designer) must consider each element of the book and what it will become in the online version (or versions).

Information that looks the same in the book can have different functions in the online help, so it's important to examine not just the appearance of the book, but also its underlying structure. The glossary mentioned earlier is a good example of this.

### **What information goes into the deliverables?**

The first step is to identify your information types. Scriptorium Publishing generally uses four fairly simple ones (described in detail in our book, *Technical Writing 101*, available through our web site); other documentation groups may use fewer or more categories. Our information types for software documentation are:

- Interface information, which describes the user interface (the information the user sees on-screen).
- Conceptual information, which describes the how and why of a particular product.
- Reference information, which provides information about the syntax of a system (for example, a list of commands and their arguments).
- Procedural information, which explains how to accomplish a particular task.

For each information type, you need to identify which deliverables should include that information. For example, conceptual information should be in the print version but not in the online help. Users want specific task information from online help and from manuals, so procedural information would go into both the online help and the hardcopy books.

In addition to various types of information, you also have to consider navigational aids. These include the tables of contents, indexes, alphabetical listings, pagination, headers and footers, and any other information that helps readers find their way through a book or help file. Navigation is generally very different from one medium to another; for example, page numbers are a necessity in a hardcopy book, but they are of little use in online help.

## A high-level planning grid

As you break down information into different categories, start looking at the high-level elements in the book and figure out what elements they will become in the help.

Table 1: High-level elements

Element	Treatment
Table of contents	Generally available in book and help. Help may include more levels to make navigation easier.
Introduction	Usually conceptual; often eliminated from help using conditional text.
Glossary	Usually alphabetical in book; often provided as pop-ups from the words inside the topics in help.
Index	Automatically hyperlinked in the online version.
Context-sensitive interface help	Often, no interface information is provided in the book.
Section	Book sections become topics in the online help.

## A paragraph-level planning grid

After assembling a grid for your high-level book elements, you can move on to paragraph, graphics, tables, and other paragraph-level elements.

In a book, you may have a Body tag that serves more than one function. For example, it might be used for body text but also for the definition of glossary terms. In a single-sourcing environment, you want a one-to-one correspondence between the paragraph styles and their function, so you would probably need to create a GlossaryDefinition tag. For a print-only book, this is unnecessary, but if you plan to convert the book to online help, you need to be able to identify glossary text uniquely.

Headings generally indicate the start of help topics, so it's a good idea to make sure that the document is broken up into reasonably-sized sections.

In a book, you may have a See Also listing at the end of a section. In the online help, this might become a “Related Topics” list, or just a list of links.

## **Choosing single-sourcing tools**

Choosing the right tool is as essential as planning a single-sourcing process. The world of single-sourcing tools can be divided into a few major groups:

- Basic tools
- Intermediate tools
- Enterprise-level tools
- Database publishing

### **Basic tools**

Basic tools require you to do a lot of tweaking to produce print and online versions. These tools are fairly inexpensive, relatively easy to use, and do not require programming ability. They also have limited functionality. They often require significant compromises in one medium or another. Tools in this category include ForeHelp, Doc2Help, RoboHelp, and others.

### **Intermediate tools**

Intermediate tools provide more features than basic tools do. Generally, they require a lot tagging and special formatting in the source files, and you need some low-level programming skills to use the conversion tool. The only strong contender in the intermediate category is the FrameMaker/WebWorks Publisher (or MIF2GO) combination. These tools are more expensive than the entry-level tools, but they are still quite reasonable. They provide better functionality and can be used to set up an automatic single-sourcing process. Strict adherence to a template is required.

### **Enterprise-level tools**

At the enterprise level, the cost of software skyrockets to \$50,000–\$100,000. These tools are appropriate only for very large organizations, and their implementation requires the services of a consultant. They can provide repositories for information and the ability to output to many media. They also provide for information reuse across an entire organization.

## Database publishing

Essentially, this is the do-it-yourself approach. The information is stored in a database, and writers create queries to extract the appropriate information and format it. Heavy-duty programming and database design skills are required.

Database publishing, although currently quite difficult, may be the direction in which single-sourcing efforts are moving.

## The FrameMaker/WebWorks Publisher approach

If you choose the FrameMaker/WebWorks Publisher combination, you'll use FrameMaker's features, such as paragraph tags, markers, and conditional text, to build in the customization that you need for a successful single-sourcing project. The following are some of the issues you need to consider as you plan your project:

- Topic depth
- Conditional text
- Context-sensitive help
- Links
- Navigation
- Chunking information

### Topic depth

Topic depth refers to the length of a section. When you create a manual or other printed material, you may not pay much attention to the length of your sections; some third-level section could be half a page, others might be several pages long.

But when you move your content online, topic depth becomes very important. Your major sections become individual topics in online help. When you determine which heading levels become individual topics, you want to make sure that you create topics that are not too long and not too short. To ensure that this works consistently, you need content with consistent section lengths.

Typically, you would analyze your files to identify sections that are too long or too short and then add or remove headings (or perhaps even add or remove content) to ensure that the topics you produce in your online help are a useful length.

## Conditional text

FrameMaker's conditional text feature lets you label information that should appear only in the printed version or only in the online version. You can set up tags such as PrintOnly and OnlineOnly. When you print the document, you hide the OnlineOnly information. When you convert through WebWorks Publisher, you remove the PrintOnly information.

A common scenario would be to tag detailed information, introductions, and conceptual information as PrintOnly. Lead-in sentences and additional navigation links would be OnlineOnly.

## Context-sensitive help

Context-sensitive help is fairly easy to implement. You can take advantage of WebWorks Publisher macros to automate the creation of the mapping file required in most help formats. To set up context-sensitive help, you create a custom marker type in your FrameMaker files, then insert markers with help IDs where the related text occurs. WebWorks Publisher uses these custom markers to create the mapping file.

## Links

FrameMaker cross-references are preserved when you convert the files via WebWorks Publisher (although the process for linking across different books is just a bit tricky). But in addition to using cross-references, you can implement additional links using FrameMaker hypertext markers. There's no difference between the links you produce with cross-references and hypertext, but there are some differences in implementation:

- Cross-references are usually faster and easier to create because FrameMaker does a lot of the work for you.
- Hypertext markers are more flexible—you don't have to use text or a paragraph number from the destination link as the hot spot.
- Hypertext markers let you create image maps.

If the formatting available from a cross-reference meets your needs, you can set up a cross-reference and save some time. If you want a more complex link, hypertext is available.

## Navigation

In online help, providing navigation is a critical element that makes your help friendlier. WebWorks Publisher lets you automatically generate links from your tables of contents and index, and also creates navigation bars on each page, generally with Next, Previous, Contents, and Index links. You can customize the navigation bars, and you can create page-level automatic tables of contents, which list the major subsections on the page.

In addition to the “standard” navigation provided by the index, table of contents, and navigation bars, you can use hypertext and cross-references to build additional navigation, such as lists of related topics. You may also want to use conditional text to hide these extra navigation features in the printed version. Another useful technique is to create pop-up links to glossary definitions, so that readers can click on the term in the help and see a pop-up definition. (Pop-up topics aren’t supported in every help technology.)

## Chunking information

*Chunking* information refers to the process of separating your content into topics. Each topic is a chunk of information. To ensure that your content works both in print and online, you want to create chunks of content that stand on their own. These chunks, or modules, of information let readers look at one chunk and get all the information that they need about that particular task or topic. Of course, you can provide links to additional information and related topics where necessary.

In a book, writers often assume that readers start at the beginning of the book and read through to the end. As a result, writers use references to other sections that are position-dependent; for example, they refer to “the previous section” or “the figure above.” These types of references are problematic even in a book because most technical documentation is not read from cover to cover—readers go to the index to find the information they need and start reading there. But in online help, the position-dependent references are completely irrelevant because readers don’t think of the online help as consisting of a string of topics with a sequence. When you chunk information for single sourcing, it’s important to eliminate all position-dependent references.

Instead of using “previous section” or “next section,” you can create references that explicitly give the topic name. For example, compare the following two links:

As discussed in the [previous section](#), you must configure the WeevilBopper before starting this installation process.

Before you begin this installation process, be sure that you’ve [configured the WeevilBopper](#).

The second example avoids the topic sequence that’s implied by the phrase “previous section.”

## Mapping FrameMaker styles to Publisher styles

The features available in WebWorks Publisher correspond to the features available in FrameMaker. WebWorks Publisher uses the FrameMaker tags, such as paragraph, character, and table tags, to determine the formatting used for the output. Basically, WebWorks Publisher replaces the FrameMaker tags (you can see these in a MIF file) with the tags need for HTML, XML, or RTF.

*Table 2: FrameMaker elements and WebWorks Publisher conversion*

<b>FrameMaker</b>	<b>How are they converted?</b>
Paragraph tags	Paragraph style definitions.
Character tags	Character style definitions.
Master pages	Thrown away. WebWorks Publisher contains its own page definitions.
Table formats	Table style definitions.
Color definitions	Thrown away, but you can specify a color in the paragraph, character, and table definitions.
Reference pages	Graphics referenced in paragraph definitions are thrown away, but you can code them into the paragraph style definition in WebWorks Publisher. TOC and other generated flows are thrown away, but the information in generated files is processed.
Variable definitions	Converted as regular text.
Cross-reference formats	Converted to hyperlinks. You can change the text of the cross-reference definition during conversion.

Table 2: FrameMaker elements and WebWorks Publisher conversion (Continued)

<b>FrameMaker</b>	<b>How are they converted?</b>
Conditional text settings	You can set what will be included or excluded based on conditional text tags. These settings can be different from the show/hide settings in the FrameMaker files.
Equations	Converted as graphics.
Graphics	The imported-by-reference graphic is copied and used, or the anchored frame is printed to PostScript and then converted to an output format, such as GIF (for the web) or BMP (for WinHelp). Unanchored graphics, graphics on the master pages, and graphics inserted by the paragraph format's Advanced properties are thrown away.
Markers	Marker style definitions.

## Contacting us

Scriptorium Publishing Services, Inc. excels at transforming complex technical ideas into clear, concise documents. Our clients, who range from start-ups to Fortune 500 companies, rely on us for the full spectrum of technical publishing services—everything from turnkey documentation to specialized technical editing and consulting. Our expert, talented staff thrives on working with challenging new technology in a deadline-driven environment.

If you have any questions about Scriptorium Publishing Services, Inc., contact:

Scriptorium Publishing Services, Inc.  
P.O. Box 12761  
Research Triangle Park, NC 27709-2761  
919-481-2701  
[sales@scriptorium.com](mailto:sales@scriptorium.com)  
<http://www.scriptorium.com>