

Practical single sourcing

Richard Truscott describes his project to set up a single-sourcing workflow using FrameMaker and WebWorks.

Like many technical authors, I find there are huge constraints on my writing time. I need to be as efficient as I can and avoid rewriting material for different purposes. This article tells how I implemented a single-sourcing solution for the Sanger Institute that enables me to:

- Reuse writing in various documentation products.
- Use the same writing for different purposes.

This article describes the method of single sourcing that I chose for my project. It is not, of course, the only way of achieving single sourcing.

What is single sourcing?

Single sourcing is a method that enables writers to develop information into modules that are reused in a range of documentation products such as:

- User manuals
- Installation manuals
- Administration manuals

- Help pages
- 'Short form' information, including quick reference guides and lists of commands
- Sales and marketing material
- Management or executive overviews.

What are the advantages?

The advantages of single sourcing are:

- Reusing text saves time by avoiding the need to write several times on the same subject for different documentation products.
- Enforcing standards and structured writing improves quality.
- Enforcing simple and consistent use of language results in a standard and even style.

Do you need to single source?

Ask the following questions to test if you need to single source:

- Do you have expert or novice users for your product that need separate documentation?

- Does your product have different levels, such as a full function and a 'lite' version?
- Will future releases of your product build on existing functionality, resulting in new documentation?
- Do you intend to produce documentation in more than one format?

Which method to choose?

A book by Kurt Ament called *Single Sourcing: building modular documentation*, which I discovered through the Cherryleaf newsletter, sets out a method for dividing content into modules, each with a single purpose. The great advantage of this method is its independence of technology; it is applicable whichever word processor or Help authoring tool you use.

Note: This book was reviewed on page 37 of the Summer 2003 Communicator.

Which software to choose?

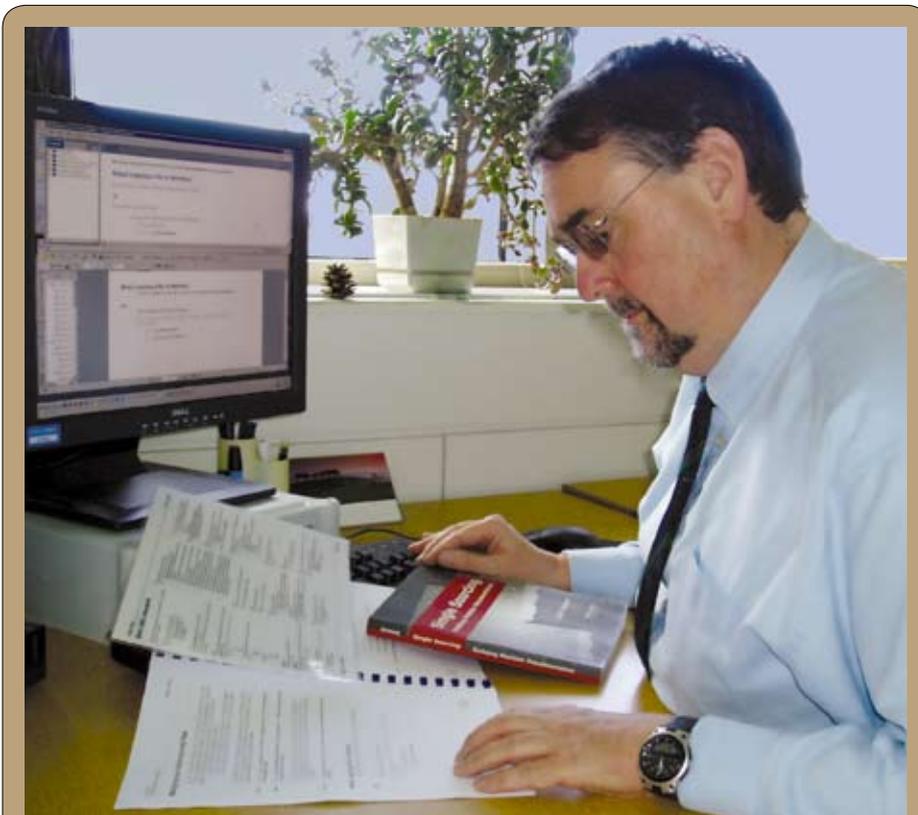
Single sourcing is not just a matter of choosing a word processor and compatible Help authoring tool. Although the need to produce Help and paper manuals is usually the reason for using single sourcing, you can use it to produce documentation in one medium only. I chose FrameMaker (version 7) for authoring because I think it is by far the best tool for producing technical documentation. It came with the standard version of WebWorks, which I used successfully even though it has limited functionality. I later purchased the full version of WebWorks 2003 Professional to gain additional functionality.

What is a module?

A module is a chunk of writing that answers a single question: who, what, why, where, when or how. Modules must be written so that they stand alone and make sense in any context; they can then be used anywhere, in any documentation product. There are two types, primary and secondary.

Primary modules

These are the main modules that stand alone or incorporate secondary



Richard Hodgkinson: 'I need to be as efficient as I can and avoid rewriting material for different purposes.'

modules. Ament suggests the following types of primary modules.

- **Definition lists.** These describe components or technologies.
- **Glossaries.** These provide descriptions for terms or acronyms.
- **Procedures.** These are step-by-step descriptions of how to do a task.
- **Processes.** These are high-level outlines of how something works.
- **Topics.** These contain descriptions, narratives, expositions or arguments.
- **Troubleshooting tips.** These explain problems and their solutions.

Secondary modules

These modules provide supporting information; they are incorporated into primary modules.

- **Examples and exercises.** These are instances to illustrate points made.
- **Itemised lists.** These are serial lists of items or sentence fragments.
- **Notes.** These small blocks of text draw attention to important positive or negative points.
- **Figures.** These include all types of illustrative material, such as photographs, drawings or charts.
- **Tables.** These present data organised into rows and columns.

Labelling modules

Ament suggests that modules must be clearly labelled to identify their type:

1. The heading has a keyword that identifies the type. For example, **Procedures** start with the keywords **How** or **To**, such as 'How to access clone requests'.
2. Filenames have a prefix that identifies the type. For example, a file containing a **Topic** has a name starting with TK. File names depend on system constraints and naming conventions already in use. Make them short, meaningful and distinctive. For example, for a topic on password rules, name the file **TC-pw_rules**.

Organising modular documentation products

Each product has some base text that contains headings and text that is particular to that product. Into this base text you import by reference the modules and figures.

In Figure 1, a user manual in A5 format and Help pages are produced from one product and a training

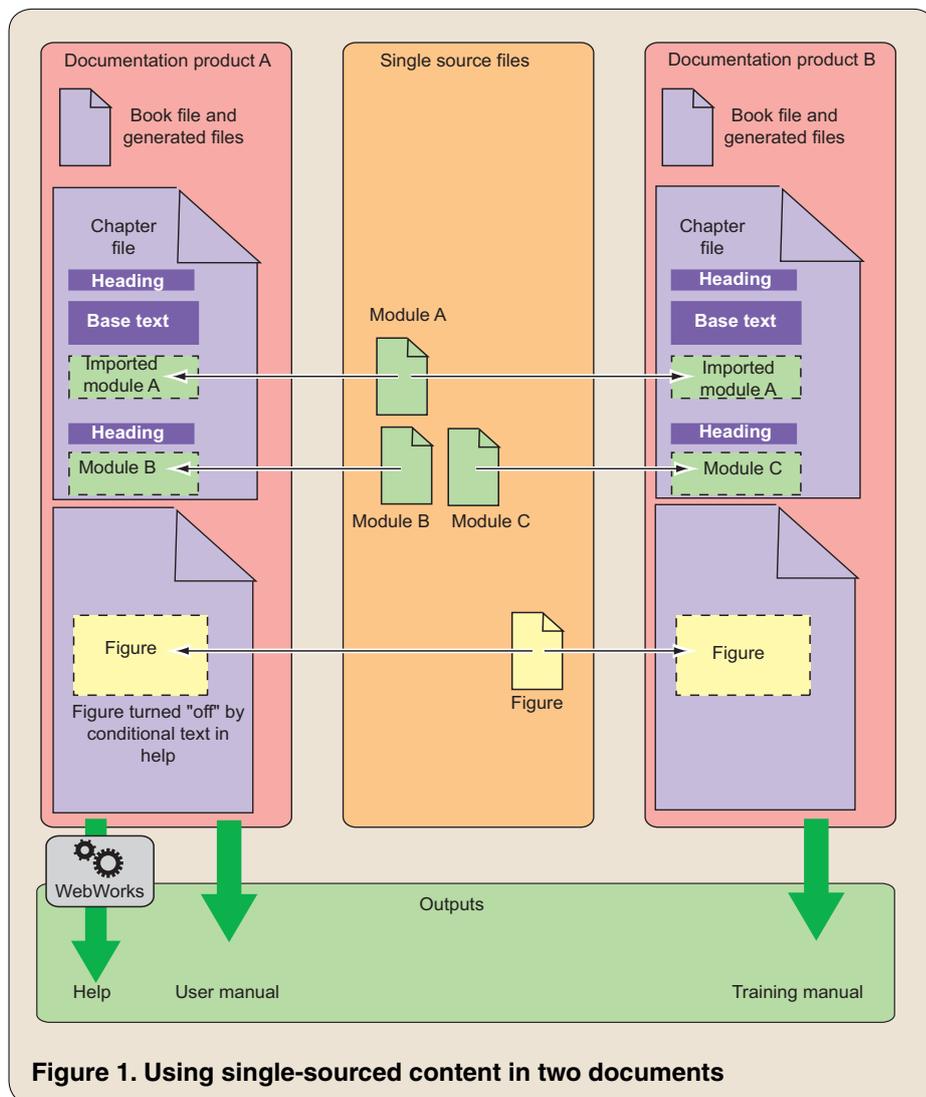


Figure 1. Using single-sourced content in two documents

manual in A4 format from a second product. Module A is used in the user and training manuals and in the Help but is not used in a get-started guide.

The user manual and Help have very similar content but without the figures in the Help, which are controlled by conditional text. The user manual has information that would be too detailed for a training manual (module B), so this module is not imported into the training manual. The training manual has additional material for information such as timetables and practical exercises and answers (module B).

What is the process for single sourcing?

An author might have either to write a new set of documentation from scratch or to convert an existing suite to modular documentation.

Preparation

Some effort and time is required to design and create templates in

FrameMaker and WebWorks.

1. **Create FrameMaker templates** for each type of documentation product. Use templates to change the format of the output, such as changing from A4 to A5. To make reusing material easier, use the same paragraph style names in all FrameMaker templates. This enables text to reformat automatically on import, without any action on your part. The paragraph styles can have different attributes (font, size, colour and so on) for different templates, but the name must be the same.
2. **Set up conditional tags** to change the content of the output, such as changing the wording between a printed or web version of the same material. For example, it might change 'This chapter describes' to 'This page describes'...
3. **Set up WebWorks.** Map FrameMaker paragraph, character and tables to the WebWorks styles.

Converting a document

To convert an existing document into modular content, start by:

1. **Identifying modules.** Work through the book chapter by chapter. Identify chunks that fall exactly into modules. Identify text that needs rewriting to make modules. On the first pass, identify primary modules; look for secondary modules later.
2. **Creating modules.** Remove modules from the document by cutting and pasting each module into a new file. Keep to the file-naming convention. Leave base text (such as headings and introductory paragraphs) in the main document; only cut and paste paragraph text. Keep to the titling rules for headings. Check each module for style and spelling.
3. **Create book files.** The text left in the book file is the base text. The modules are now in separate files. Import the modules into the main document (File>Import>File in FrameMaker).

Creating a document

To create a new modular documentation project, start by:

1. **Create books.** Set up an empty book for each documentation product. Add any standard text (for example, copyright). Set up the files for generated text (such as the table of contents and index).
2. **Create base text.** Put in paragraph headings and introductory material. This becomes the base text. What you have created is a paragraph plan.
3. **Creating modules.** Write the modules to fill the documents. Spell check modules as you create them. Keep to the file-naming convention. Import all formats from the book file.
4. **Cross-reference and index.** Set up cross-references to headings only (see *Cross-referencing modules*).

Producing a document

To continue your converted or new modular documentation project:

1. **Create printed and PDF outputs.** Create new outputs for different purposes. Use conditional text to control content.
2. **Create Help.** Use WebWorks to

create Help pages. Use conditional text to control content.

3. **Edit books and modules.** Review and test the book. Make changes to base text and modules. Check spelling both in the book and in individual modules.
Warning: FrameMaker ignores the modules when checking the book.
4. **Edit Help.** Review and test the Help. Make changes to base text and modules.
5. **Publish.** Publish Help to web servers. Publish PDF files to document repositories. Print paper copies.

Cross-referencing modules

The style adopted by the Sanger Institute is to have:

- A table of contents at the beginning of each chapter, normally listing level-one headings. This improves navigation and provides a taster of what is to come for readers of the paper version. I set up WebWorks to create a new page at each level-one heading, so the chapter-level tables of contents are useful to help readers as navigation tools. WebWorks also provides 'breadcrumbs', index and search as navigation aids.
- 'Return to top' links to the end of each major section. These are conditional and appear only in the web version.
- No cross-references from base text to imported modules; FrameMaker has difficulty updating such cross-references, although cross-references within modules and between base texts work correctly.

Organising the files

A logical file structure is required to keep track of the folders and files created in a single-sourcing project. The choice may be constrained by the IT system you use.

At the Sanger Institute, I work with Microsoft Windows XP where the '/My Documents' folder is stored on a networked drive that is backed up every day. I create illustrations using Adobe Illustrator and store them in a central library on the network drive. The structure contains:

1. A **Project** folder containing everything for the project except the glossary modules, which are common to more than one project.
2. An **Images** folder for the whole project. The images, which are in

enhanced metafile format (EMF), are 'imported by reference' into the FrameMaker files.

3. A **Modules** folder to keep all the modules for the project.
4. A **Book** file for each document product.
5. A **WebWorks** folder for all the files generated by WebWorks.
6. A **Glossary** folder for all products. 

Resources

Cherryleaf newsletter
www.cherryleaf.com

FrameMaker
www.adobe.co.uk

WebWorks Publisher for FrameMaker
www.quadralay.com

References

Ament, Kurt (2003) *Single Sourcing: building modular documentation*, William Andrew Publishing, ISBN 0-8155-1491-3.

Darley, Justin (2003) 'Implementing single sourcing', *Communicator*, Spring 2003 and Summer 2003, ISSN 0308-6925.

Redish, Janice C (2000) 'What is information design?', *Technical Communication*, vol 47, no 2, Society for Technical Communication.

Rockley, Ann (2001) 'The impact of single sourcing and technology', *Technical Communication*, vol 48, no 2, Society for Technical Communication.

Various authors and titles (2003) *Technical Communication*, vol 50, no 3. Society for Technical Communication.

Richard Truscott MISTC MSTC MCM1

is the sole technical author working at the Sanger Institute, where he writes computer software, operations and robotics manuals, online Help and online manuals. He has worked as a technical author for the last 10 years, having previously worked in telecommunications for BT, Nortel Networks and Fujitsu Telecommunications. E-mail: rjt@sanger.ac.uk

The Wellcome Trust Sanger Institute is one of the leading genomics centres in the world, dedicated to analysing and understanding genomes. Through large-scale analysis, focused research and collaborations, the Sanger Institute's programmes underpin biological and medical research worldwide.

Website: www.sanger.ac.uk