



# Beyond the Buzzword: Single Sourcing

Kazu Nitta

BY SEAN BRIERLEY, *Senior Member, Central Connecticut Chapter*

Single sourcing, which has been a buzzword in technical communication for several years, is now emerging as a practical, efficient, and cost-effective method for creating multiple deliverables. You might want to consider single sourcing if you have some documentation projects that repeat, if not every one of your projects is a one-off, if you reuse some of your content, and if your budget planning includes more than just the very next project.

This article is intended to raise the awareness of publications managers and technical writers in general about the utility and applicability of single sourcing to a broad range of document types. I'll also discuss some of the workflow issues that need to be considered before implementing single sourcing.

## Why Single Source?

Single sourcing is a documentation workflow that creates multiple deliverables from one unmodified source document—that is, in the process of creating the deliverables, the source document is not edited or modified. The deliverables produced by single sourcing can differ not only in form—for example, online help and print-ready documentation—but also in content. Thanks to this flexibility, different content can be provided to different audiences, so that (for example) novice, intermediate, and advanced users of a product can receive only the information that will be helpful to them.

If you produce multiple deliverables that share some of the same content, single sourcing can reduce the resources in time and staff you need to produce them. Single sourcing really begins to shine when it's

time to edit and update the deliverables, because all the changes can be made to a single document. With traditional workflows, changes would have to be synchronized across many source documents.

Single sourcing should not be confused with *repurposing*, which involves re-tasking an existing document designed for one deliverable, output, or audience and using it for a different deliverable, output, or audience. Repurposing is useful for turning around a one-off project, while single sourcing is designed to repeatedly create multiple deliverables from one source. It is equally suited to highly structured companies, which profit from its template-driven workflow or reliance on a complex database, and to highly informal companies, which benefit from the short development time needed to produce multiple deliverables.

### Single-Sourcing Toolsets

Single-sourcing toolsets can be loosely classified into two categories: high-end and low-end. They include authoring software, such as Microsoft *Word* and Adobe *FrameMaker*, in addition to the single-sourcing tools themselves, which interact with the authoring tools to create multiple deliverables from a source document. High-end tools offer more flexibility and accept more types of documents, while low-end tools typically adhere to a traditional book and online help model.

Low-end tools for single sourcing include solutions that use template-driven workflows, such as *FrameMaker*-based Quadralay *WebWorks Publisher Pro-*

*fessional* ([www.webworks.com](http://www.webworks.com)), Omni Systems' *Mif2Go* ([www.omsys.com](http://www.omsys.com)), and Microsoft Word-based *AuthorIT* ([www.author-it.com](http://www.author-it.com)). High-end tools include those that feature XML rendering of content from databases, such as Arbortext *Epic* ([www.arbortext.com](http://www.arbortext.com)) and *FrameMaker*-based *Miramo* ([www.miramo.com](http://www.miramo.com)). Such tools are suited to large companies and publications departments.

Tools that have traditionally been used to create separate online help, such as ForeFront's *ForeHelp* ([www.forehelp.com](http://www.forehelp.com)) and eHelp's *RoboHelp* ([www.e-help.com](http://www.e-help.com)), might also be considered for single sourcing at companies that manufacture software. However, these tools currently work

best in a traditional multi-source environment in which you might begin with a single document but end up supporting two or more sources.

Like all software, single-source toolsets have different capabilities that should factor into your selection. For example, *FrameMaker* includes conditional text, a feature that lets you define specific content for a specific deliverable. *AuthorIT* boasts some content management and security features, which are important if your structure requires that different authors have different access rights to content. Some tools are capable of accepting a variety of authoring-tool file types as input, whereas other tools, like *WebWorks Publisher*, accept as few as one authoring-tool file type. Toolsets also have varying capabilities for accepting or accommodating customizations and the variety of customizations that are permitted. Another consideration in selecting a toolset is the ease with which it permits the automation of the single-source workflow, a key consideration in reducing the time needed to finish a project.

Whether any of these single-sourcing tools fits within your budget depends on your corporate needs and philosophy, but the benefits of single sourcing in reducing the staff hours needed to create your deliverables will more than pay for these tools in the first or second project. For more information on single-source toolsets, refer to the resources listed at the end of this article.

### Implementing Single Sourcing

The key to implementing single sourcing is determining what you want for deliverables in terms of content and form, and working to the strengths of your toolset to create and adhere closely to templates. With an understanding of the role of templates in single sourcing, you'll be better able to meet the requirements of moving to a single-sourcing workflow.

The templates used for single sourcing usually exist both in the authoring tools and the single-sourcing tool you choose. The purpose of the template in the authoring tool would be to define and tag the content that you author, while the purpose of the template in the single-



A small software manufacturer distributed online help and printed documentation using a traditional multi-source workflow. The printed reference book was written first, and the online help was written based on the printed documentation. The time needed to produce the online help was eighty hours, in addition to the time needed to produce the printed document. That company now turns around equivalent online help in three hours through the use of single sourcing. The savings is seventy-seven hours per online help project, of which there are eight per year, for a net total of 616 hours saved per year.

The costs for switching to a single-sourcing workflow included the purchase of an \$800 single-sourcing tool, sixty hours of template design (including testing and redesign based on feedback and familiarization with the new tool), and \$2,000 and forty hours in training, including travel. The commitment of 100 hours and \$2,800 did not pay for itself in the first project, but by the second or third project these expenses had been recovered. An additional benefit that didn't show up on the ledger sheet was the goodwill generated by training the staff on a useful and emerging workflow. Finally, twenty-four hours of downtime were needed to establish the preliminary templates. The rest of the template and workflow development was done on a live project, and the formal training was undertaken six months after the workflow had been established, to reinforce and hone the skills developed while implementing the new workflow.

sourcing tool would be to manipulate the content for the various deliverables you produce. Such templates can also contain modifications to the content, so you can manipulate the content for a specific deliverable. The goal is to create templates that let you completely automate the process of moving content from a source document to the deliverables. Implementing single-sourcing templates should be no more time-consuming than the process for any tool or workflow, and the payoff is potentially huge in bringing well-designed deliverables to market more quickly and efficiently.

After you define your templates, you need to map them between your single-sourcing and authoring tools and incorporate any customizations needed to create multiple deliverables automatically. Because information is taken from a source document and mapped to your output, it is a good idea to carefully consider content-based templates as well as formatting-based ones. That is, consider defining your templates according to the content they contain. Knowing that a certain style contains a certain kind of content makes it easier to design your templates because you know whether the style needs to appear in all deliverables. For example, you might decide that a style called “Callout” is Arial 9 point bold and that it contains non-essential information designed to reinforce something elsewhere in the text. With this definition in mind, you can decide to omit the style from an online help deliverable but to include it in a printed deliverable. Indeed, the formatting of a style in single sourcing is largely irrelevant because it often changes based on the deliverable in which the style appears. It is preferable to define styles according to content, because content remains constant across the deliverable set.

Additionally, your content has to be chunked so that it works in each deliverable. For example, printed documentation, or PDF, flows from page to page, so readers experience continuity outside the section in which they are working. But navigation through online help has an element of random access. If PDF and online help (or HTML) are two of the deliverables you are authoring, your tem-

plates and content mapping have to account for how readers will use each medium. You might map headings to new HTML pages for the online help system, so that the material is chunked into manageable sizes for just-in-time information delivery. However, if your templates permit heading styles to be followed immediately by other heading styles in your single-source document, you will create blank HTML pages comprising only a heading.

Another issue when implementing single sourcing concerns navigability within HTML or an online help project. In a printed document, you can find related information by thumbing a few pages before or after the section you are reading, but in online help there is no before or after. As a result, you might consider designing your styles so that links to related information, such as all the subheadings under a second-level heading, appear in a particular section. For example, an HTML page that you output might contain a bulleted list of related links, each of which might be level-three subheadings in a printed deliverable or PDF. You can choose whether to include such links in all of your deliverables or in only one kind of deliverable.

Deciding whether to display certain types of links in certain kinds of deliverables raises the question of displaying or hiding certain content from deliverables—that is, creating from one source multiple deliverables that differ in content. For example, you might design a single-source document that addresses the needs of three separate audiences, such as novice, intermediate, and advanced users of a product. By assigning novice, intermediate, and advanced content to different tags, styles, or conditions (depending on how your tools handle these things), you can create multiple deliverables from one source just by choosing which tags to output.

In fact, there are many options and directions in which you can take your single-source workflow, limited mainly by your own training, education, experience, imagination, and the tools you have chosen. Implementing single sourcing is the same as implementing any workflow: To succeed you must be deliberate and

thoughtful about what you are doing. You have to consider your deliverables beforehand, understand why your current workflow delivers what it does, determine what you want to deliver from your new single-sourcing workflow, calculate what needs to be incorporated into your single-sourcing templates, and use your single-sourcing tool to create those mappings, conversions, and customizations automatically.

### The Bottom Line

Single sourcing is a workflow that addresses the resource needs of the authoring company while producing multiple document deliverables, each of which effectively meets the needs of the audience. Implementing a single-sourcing workflow requires careful consideration of the tools, templates, and other elements that you need to use. However, if you produce multiple deliverables that share some content while differing in output type and/or target audience, you can use single sourcing to greatly improve the efficiency of your workflow. The bottom line is that you should consider single sourcing as an efficient way to save time and money in delivering a useful set of documents to your customer. **i**

### RESOURCES

[groups.yahoo.com/group/authorit-users](http://groups.yahoo.com/group/authorit-users): The *AuthorIT* user group.

[groups.yahoo.com/group/HATT](http://groups.yahoo.com/group/HATT): The Help Authoring Tools and Techniques group, a tool-independent peer-to-peer group.

[groups.yahoo.com/group/wwp-users](http://groups.yahoo.com/group/wwp-users): The *WebWorks Publisher* user group.

[www.adobeforums.com](http://www.adobeforums.com): An official Adobe product forum site, including *FrameMaker* and *FrameMaker+SGML* forums.

[www.frameusers.com](http://www.frameusers.com): An Adobe *FrameMaker* online user group, list, and resource.

**EDITOR'S NOTE:** STC now has a Single Sourcing SIG. For more information, please visit [www.stc.org/SIGS/single\\_sourcing.html](http://www.stc.org/SIGS/single_sourcing.html).