

Definition of Terms



Content Reuse:

We all reuse content all the time. Any time you cut and paste from one document to another, any time you output a document in both word and PDF, you are reusing existing content. And that's a good thing—better to reuse than recreate, but it's not single sourcing.

Single Source:

Single Sourcing is a specific methodology to manage and sequence content and maximize its reuse. To do this, regardless of what platform or authoring tools you are using, you identify and categorize chunks of content, store and manage these chunks, and output them in multiple formats. So as opposed to cutting and pasting to and from multiple sources, you have one, customizable data source that can be used in multiple outputs. The basic content remains the same, the output varies. And most important, updates are automated. That is, the system manages the content outputs, so that if you update a chunk it is automatically updated everywhere it appears.

DITA—Darwin Information Type Architecture

DITA an accepted standard designed specifically for tagging, storing, and managing single source content. DITA is an OASIS standard. OASIS is a standards organization that was formed in 1983 to manage SGML, Standard General Markup Language. The “ML” part of these words, the Markup Language indicates the ability to tag text, to create a label to define attributes of the text, such as Title, Section Title, Body, Bulleted List. (HTML is HyperText Markup Language that determines web page format, and XML is Extensible Markup Language). IBM spearheaded DITA development starting in 2001 and OASIS adopted DITA as standard in 2005. The DITA standard is compatible with robust authoring tools like FrameMaker, for example. DITA uses XML, which is a linear descendent of SGML. XML is a set of rules for

encoding text strings. By now DITA is a mature environment with lots of vendor add-ons that enhance its abilities, and it is open source. It also has the added benefit of allowing you to import code strings for parameter tables, for example. What makes it magic is that once you've encoded the data, it can be processed by a computer—and thus automated. This is how you get that wonderful benefit of customizing output, and automating content updates to all outputs.

Information Delivery is Changing



The model for Single-Source data fits the current trend that people search for information online, as opposed to reading it in a book. Online information can be devoid of both context and sequence. Each chunk of content must be able to make sense on its own. The detachment of data from context and sequence has made single sourcing not just important but essential to satisfying users. They may never see the book—they'll judge the quality of the information by the quality of the parts, and if those parts are contradictory, or inaccurate, you have a big credibility problem. Single sourcing is designed to solve these problems by ensuring that you have accurate, reviewed content for every topic.

The Single Source Model

A true single source environment is a collection of topics—they may be as small as a title or as long as a detailed procedure, but they stand on their own. They are stored once—one instance only—they may be customized for different outputs, but there is one approved content chunk. They are stored in a single repository and their output is customizable and updates are automatically tracked and managed by the environment—if you make a change to a content chunk, it is updated everywhere that chunk is used, any time it is output. A content chunk by the way doesn't have to be text—this is especially important for those of you creating training. You can store avi and wave content as well as text, and Single-Source training tools are burgeoning.

The Single-Source Process

1. First, you gather content.
2. Next you analyze the data, to identify logical chunks of information. These are the pieces that stand alone. They may be overviews, procedures, reference items, etc.
3. Once you define content chunks, you validate them—this means that instead of giving your technical experts a manual to review,

they only review the specific content relevant to them. Once the relevant individuals approve their content, it's validated—no more reviews.

4. You store chunks in a data repository.
5. Next, you determine output types, and create maps and “transforms” that reflect this design. Maps are DITA instructions as to content and sequence; transforms are similar to templates that define output.
6. Once this process is complete, any change you make to the content updates to all its outputs.
7. In addition publishing content in different media, you can also customize output for different job functions—one view for developers for example, another for finance, or support or training.
8. And more than that, you can customize output based on experience levels, internal or external customers—the ability to customize is not limited, and represents only minimal additional overhead.



You can see how this vastly streamlines the content review and update process. Review comes at the content level, not after content is assembled.

Because XML is a machine-readable language, automation can happen at several levels. You can import code strings, such as parameters and reference tables. And you can customize and automate output based on XML tags.

Benefits of the Single-Source Process

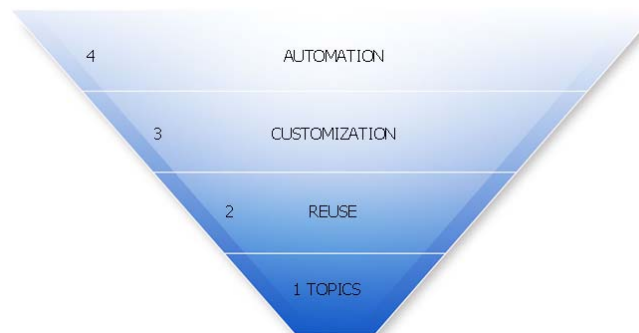
- **Increases accuracy and consistency**—data is validated once, and accurate, consistent data is propagated wherever needed. You can import parameters and code strings directly from code, eliminating errors of transmission.



- **Streamlines review cycle**—Subject Matter Experts can focus on their content chunks to ensure accuracy; once validated they do not need further review.
- **Allows for customizable output**—Once you create the content, you can customize by media type, delivery channel, level of expertise; any need you have for output can be easily accommodated.
- **Simplifies translation**—Instead of sending deliverables to translation, you send the XML source files. Because you can better manage the translation workflow, you can send content for translation as soon as it is complete, regardless of the deliverables in which it will appear, as well as send only those files that contain changed content. This flexibility allows you to better control costs and reduce release delays due to translation time. For example, if you have a product guide that uses 200 topics (125 pre-existing topics and 75 new topics), you send the 75 new topics to translation instead of incurring processing costs on all 200 topics. In this case, the savings is over 60%.

Stages of the Single-Source Process

You don't need to revolutionize your processes to start preparing for and reaping the benefits of single sourcing. This graphic shows four stages of single sourcing, and the widening of the funnel indicates the increased ROI at each stage:



- 1) Level one requires no tools at all—that is simply identifying topics—content that you reuse in your organization. Simply catalog and validate key topics that are used in multiple places (such as corporate history, support phone numbers, product overviews, procedures and so on). This is the first step. Once you have the validated best corporate history or product overview, you make that the standard set of paragraphs everyone uses. Standardizing models and templates is also beneficial. You can cut and paste at this first stage.
- 2) The next big step is to move to structured reuse. You move content into a data repository, and rather than cut and paste, you define standard output, and reuse the data for the output. You don't add new, customized outputs, but you change the way you produce deliverables.

- 3) The next phase is to make your documents fit your users. This was too expensive in our old book model, but customizing what different users see adds the benefit of increased user satisfaction.
- 4) Finally, at level four, your systems are all in place and you start automating the process through the tools available in your standard environment, and at this point you are producing quality content with minimal effort customized to specific needs—it's a huge win.

How Do We Start?



Needs analysis:
What have we got?
Who uses it?

Moving towards a Single-Source model doesn't have to be intimidating. The first step should be familiar to most of you—a needs analysis.

You need to determine what content you have, where it's duplicated, and how people use it. You need a detailed analysis of your deliverables themselves, what tools people are using to create them, where the content comes from, and where it overlaps. You need to know what internal and external users like about your current materials. What do they use the most or find most helpful? What do they complain about? What would they like to have that they don't?

Once you have that information, you need to analyze one more thing: the current cost to produce your materials. This includes writing, editing, reviewing, and publishing. Who's involved? How many subject matter expert hours does it take? You need to measure this cost or you can't measure Return On Investment (ROI).

When you have a cost basis, you can project savings with a Single-Source solution. Remember that you have to allow for the initial setup of the new environment, and most of your savings will come after that initial investment—with subsequent releases, because you need to invest time and money in changing the process and doing it right. A rule of thumb is that if you can single source one-third to one-half of your content, there is a substantial ROI—the more the better.

And finally, pick a small project to start with, so that you can be sure to succeed and get buy-in. Start with a project with lots of reuse, lots of inaccuracies, and you can show dramatic improvement right away.



**Start small
and succeed**

Questions?

We're here to help: www.techprose.com, 800-903-4330

