



# simply written, inc.

## An Introduction to Single-Sourcing

*..a bit about what it is and some pros & cons to  
think about*

---

### Information Summary

---

#### get in touch

Although located in the midwestern portion of the United States, we can work with you wherever you're located, thanks to the wonders of modern technology.

#### by U.S. mail

8522 east 56th street  
indianapolis, in 46216-2024 usa

#### by telephone

317.562.9298

#### by email

jerilynn sander, ceo/information architect  
jsander@simplywritten.com

#### on the web

[www.simplywritten.com](http://www.simplywritten.com)



# contents

---

<b>the current approach</b>	<b>3</b>
there are external documents	3
and there are internal documents	4
when information changes	4
what's wrong with my current approach	5

---

<b>a reality check</b>	<b>5</b>
living documents and too much technology	5
so what's the solution	5

---

<b>so what is this single-sourcing</b>	<b>6</b>
how about a definition	6
maybe an example	8
a few other terms	8

---

<b>which is the "right" way</b>	<b>9</b>
when to move to single-sourcing	9
pros	10
cons	10

---

<b>single-sourcing: a quick &amp; easy approach</b>	<b>11</b>
pros	11
cons	11

---

<b>single-sourcing: a middle way</b>	<b>12</b>
with the analysis done...	13
pros	13
cons	13

---

<b>single-sourcing: the high end</b>	<b>14</b>
pros	15
cons	15



This document provides an introduction to the concept called “single-sourcing” – a more efficient and less expensive way to design, create, maintain and produce the overwhelming amount of content (words and pictures) we use to describe products and services, rules and regulations.

Before you can evaluate whether single-sourcing could be helpful to your organization, let’s review how documents are currently created, then we’ll put a definition on the table, and look at some of the pros and cons to single-sourcing.



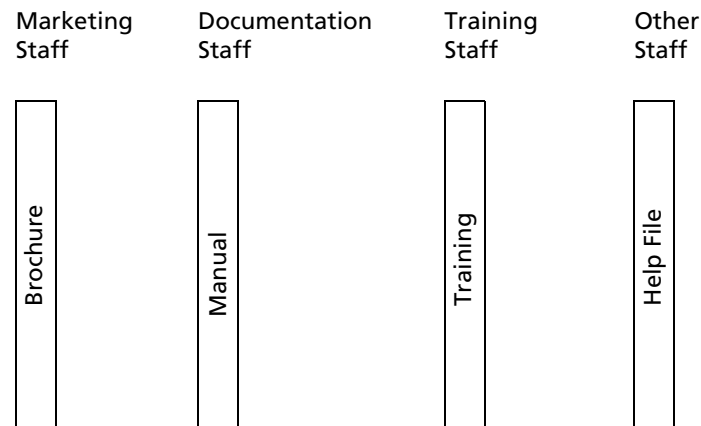
## the current approach

### there are external documents

In most organizations different people create a specific document to serve a specific purpose. For example

- the marketing department creates brochures that are visually appealing and help sell something
- the training department creates documents that help people learn to do something
- the hotline creates documents to help them answer questions they hear every day
- the programmers create help files that provide on-the-spot answers when your customer is using tool x
- someone else creates manuals (which, by the way, aren't used anyway, right?) to use as a look-up tool if someone has a question later on down the road

So the development process is like a pipeline or tunnel, each group creating what they need to get their own job done. For example, let's say that your company comes up with Awesome Tool 6000. You might have the following



And the approach works just fine, thank you very much. We're selling lots and lots of this great tool and life couldn't be better. We don't want the documents to be clones of each other – otherwise we wouldn't need different documents. Each serves a different purpose. The information isn't the same, the needs are different.

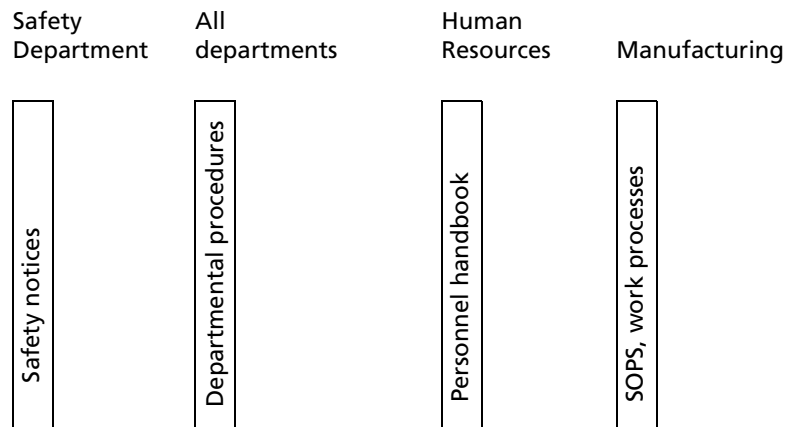


**and there are internal documents**

Or maybe you don't have to worry about external documents since your job is to worry about internal documents such as

- safety notices
- standard operating procedures (SOPs)
- benefit handbooks
- memos and letters
- work processes

This development approach is also like a pipeline or tunnel, each department creating independent documents that meet their needs. You might have the following



**when information changes**

So what happens when something changes? A new law is passed? The regulatory requirements are different? There's a new product description? The product name changes from Awesome Tool 6000 to Awesomely Kewl Tool 9000?

It depends.

The folks in one group may, or may not, talk to the folks in another group about what they're doing with their documents. Training *may* hand their file over to the documentation group to use as source material. Marketing *may* send a copy of the latest product description to all of the other groups. Department A *may* tell Department B that their needs have changed.

Or not. It just depends, on a variety of factors – time, money, understanding, internal processes, departmental rivalries. And all of the documents may, or may not, get updated with the latest and greatest details. It just depends. Because, quite frankly, many organizations don't have a process in place for handling these details.



---

### what's wrong with my current approach

Maybe nothing. You may be doing just fine and don't even need to think about how content (the actual words, punctuation and grammar used as the source material that explain, describe, and support products, services, rules and regulations) gets from your organization to those who need to see or use this information.

Maybe something, since the old copy/cut/paste approach doesn't work so well when you're not even sure where everything is being used, or will be used in the next 6 weeks or so! We simply can't remember where each individual phrase was used or where the latest product description is stored or where the information might be needed tomorrow. And the traditional document development approach just wasn't built to handle all of this constant updating and multiple deliverables.

## a reality check

---

### living documents and too much technology

The reality is documents are "living". The content changes, is needed elsewhere, needs a face-lift. No big surprise there.

*And* the reality is that technology is driving the bus these days.

*And* the reality is that we're producing more information and putting it more places than ever before, thanks to continually evolving technological changes like the Web. We're expected to have all sorts of information available upon request. We now "need" stuff in a help file *and* on our website *and* in our marketing brochure. And we need to customize the information so that it will work for that great new business partner, while maintaining the current version that doesn't reference that partner.

Oh, and by the way, we're supposed to get all of this done with budget cutbacks and hiring freezes.

---

### so what's the solution

The solution is to do things differently by

- using a well-defined process for creating, updating, and maintaining your valuable information assets
- creating reusable content "chunks" or "objects" instead of individual documents
- analyzing your needs and building a plan that helps you keep up with your requirements
- defining rules for all parts of the process and insisting that everyone, *everyone*, stick to the rules
- using all that grand new technology to make managing this whole thing easier and more efficient



## so what is this single-sourcing

It's a way to do things differently that will both help you and drive you crazy with all of the details you need to attend to. With this approach you can reduce the time-to-market and the costs associated with developing content by 30-60%. And no, those figures aren't fictitious. You truly can save that much. However, what you save depends on a lot of factors, the most important one being your ability to plan first.

---

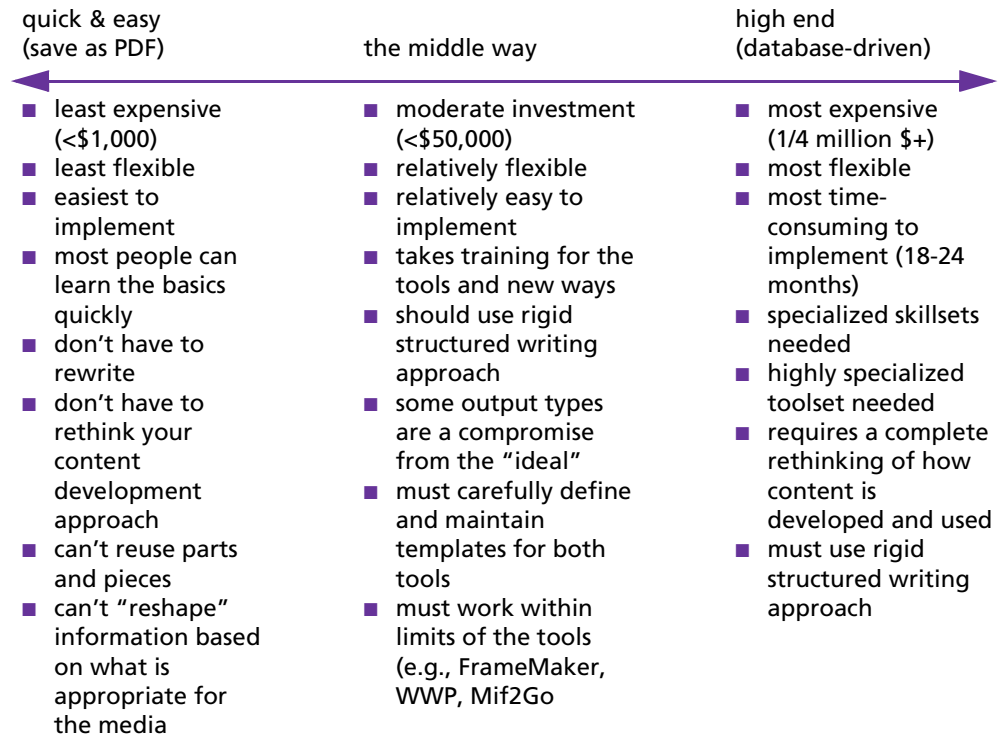
### how about a definition

So what is "single-sourcing"? It is an updated approach to creating and maintaining that valuable asset called content or information or documents or whatever you happen to call it. It is using a well-developed document manufacturing process instead of throwing stuff on a static page and sending it out the door. So instead of the "pipeline" approach where you create a document, you create containers of information that are shared, or not shared, as needed.

But before you get too excited, you need to know something: there is no one "right answer" to this question or even one definition. Some folks will tell you that the only way to single-source is to use a database, others will tell you that they're single-sourcing because they're creating PDF files from Microsoft Word and putting the PDF file on a website and a CD-ROM.

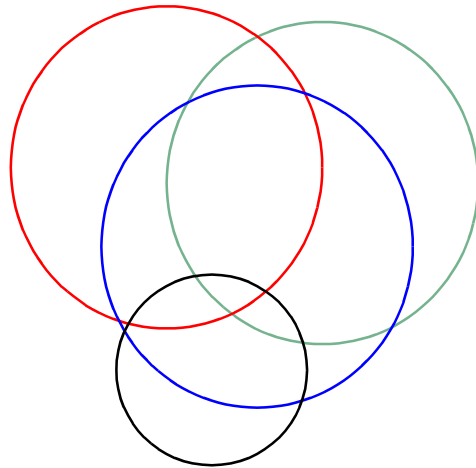


In our opinion, they're both forms of single-sourcing. So how about we compromise and look at it as a *continuum* of solutions rather than one single solution, each with pros and cons to be considered as you look at the needs of your own organization. (We use *write once, use many*, which pretty much covers all the bases.)





**maybe an example** To continue with the example of the Awesome Tool 6000...after digging into your content you find that you *do* have some information overlap in the different information products: all of them include a product description and 3 of the 4 include step-by-step procedures. So now you realize that your content looks more like overlapping circles than parallel pipelines.



With a single-sourcing approach, you would write and store the procedures in one place and use the same text in each of the 3 places it's needed. Now when it comes time to update the information, you change it in a central location (the "parent" or "source") and all of the other places where that information is used ("the children") get updated! Numerically, you've reduced your costs by 60%. In reality, it is probably closer to 40% since there are other tasks you have to deal with.

---

**a few other terms** Before you read on, you might want to know a few more terms (for additional terms, please [click here](#) to see the glossary we keep on our website or go to <http://www.simplywritten.com/library.htm> if you're reading a printed copy of this document).

- content – the actual words, punctuation and grammar used as the source material that explain, describe, and support products, services, rules and regulations
- content-based structure – a way of organizing information based on the similarities in the information
- information product – a specific group of content assembled to meet the needs of a specific group of End-Users for a specific purpose



## which is the “right” way

There are many different ways to single-source, and the right way depends on what your organization needs. Single-sourcing is a combination of planning, writing, and tools that lets you create a single place to store content, a repository if you will, then use that repository to create a variety of different information products (or what we used to call “documents”).

If you took a look at the continuum, you saw that there are dramatically different costs involved with each solution. So you’ll also have to decide how much a new solution is worth to you.

Ideally, you’re going to doing single-sourcing with an awesome database, content management tools, and a lot of other tools that let you extract and reshape content on an as-needed basis.

In reality, many companies can’t afford the \$250,000 + financial investment and the 18-24 month implementation cycle to do true single-sourcing. So you can start with a phased approach and something that may be less than ideal, but still better than what you’re doing right now.

---

### when to move to single-sourcing

You may need to move single-sourcing if you need to

- produce multiple deliverables with the same, or similar, information (e.g., a printed document and website)
- create content for products that are similar, say the Awesome Tool 6000 Model A and Model B (especially if there is at least a 40% content overlap)
- share content between departments or groups
- share content for different End-Users (e.g., managers, sales reps, administrators)
- update your content frequently
- translate the same information into multiple languages
- provide similar content to multiple legal entities (e.g., regulatory agencies in different countries)

Like everything in life, you have to give something to get something, so let’s look at some of the trade-offs of single-sourcing.



---

**pros**

- You only write information that is reused in multiple places once – not several times.  
That means no more parallel development within your own group, wondering who has the latest and greatest information, wondering if you found all the places a particular piece of information was used. If you can get multiple groups to work together, you don't have parallel development *between* groups.
- You only have one review process and one place to make changes.
- You can work more efficiently and get more done with fewer resources.
- You can make changes in one place and know that all of the information products are “instantly” updated.
- You can provide more types of information to more people without substantially increasing the overhead costs.
- You know that your information is accurate.
- You can predict what is needed and use technology to assemble information products on an as-needed basis.
- You analyze everything before you begin working so you have little, if any, need to keep going over and over the same information, wondering where to put this or that.
- You can save 30-60% on the cost of developing information products.
- You develop a document that guides the development efforts, so everything is pre-planned and ready to be assembled.
- You can teach people how to do what needs done because the structure is already built.

---

**cons**

- It can be expensive to implement.
- There is a 99% chance you'll have to bring in new tools and techniques.
- To get the best ROI out of a high-end system, the solution should be cross-departmental, and that brings a whole slew of new challenges to the table.
- You have to be willing to compromise since you won't always be optimizing for a specific deliverable any more.
- You have to stop and plan, everything, before developing content.  
Unless you're implementing the simplest single-sourcing solution, you can no longer “grow” documents the way a reef grows. No more last minute “Oh, let's just add this interesting tidbit”. You'll start needing templates, styles, and even more advanced gizmos. Not every company is ready for this kind of regimentation, but without it you're wasting your time moving to an advanced single-sourcing solution.



- Many content developers don't like being told what to do, what goes where, and how they "have" to do things.
- In general, people resist change so the content management challenges can derail a single-sourcing project more quickly than anything else.
- Except for the simplest solutions, it takes approximately 3-6 months to implement a medium level solution and can take 18-24 months (or more!) to implement more sophisticated solutions.

## single-sourcing: a quick & easy approach

The simplest scenario might be something like this. You have a 10-page manual in Microsoft Word format that needs to be on paper, on a CD-ROM and on your website.

You purchase Acrobat, load it on your system. Since Acrobat obligingly puts a nice little icon on the toolbar, you click that button and you have a PDF (portable document format) file that can be attached to an email, stored on a CD-ROM, and can be accessed as a downloadable file from your website.

---

### pros

This solution is fast, inexpensive and easy. You don't have to rewrite anything, do much planning, or change the way you write content.

---

### cons


You can't do anything else with the content: it's wedded to a single file with a specific format. You can't

- break the file into reusable chunks that can be used somewhere else
- update one piece of information and have that change automatically made somewhere else
- change the look (format) to accommodate the unique needs of different deliverables



## single-sourcing: a middle way

At this level, there are more ambitious needs. You may want to reuse just portions of the content or take advantage of special formatting or navigational capabilities of different media or have specific content translated into another language. The existing information products (formerly known as “documents”) may be needed for different End-Users or be longer or more detailed or include a larger variety of information (e.g., procedures, contextual information, references).

 This happens to be the area we we’ve focused our efforts. However, there are other consultants who have focused on the high end solutions. *by the way* If you need information about those consultants, let us know.

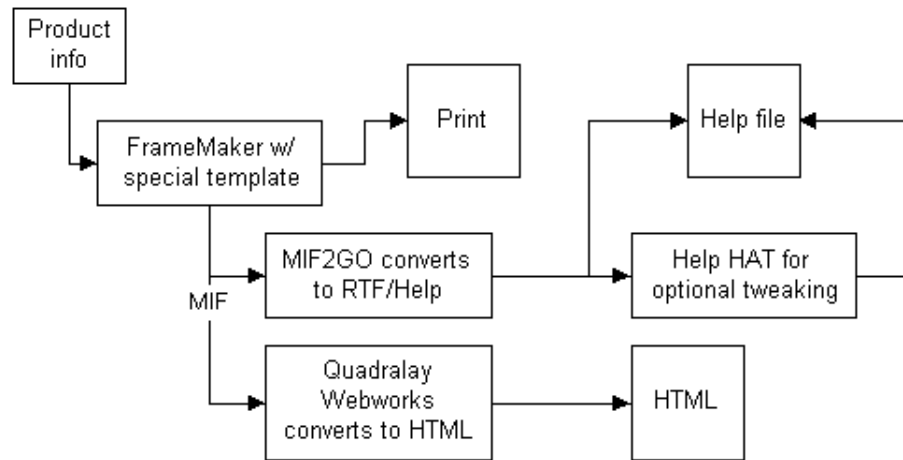
To do the middle way, you

- 1 *Must* start with an analysis of
  - ◆ your End-Users
  - ◆ communications goals
  - ◆ information products
  - ◆ Delivery Methods
  - ◆ ...anything else that can be nailed down
- 2 Pick the tools that will support the variety of needs that you have (as of this publication date, there aren’t a huge number of them that fall in this range). Some of them include
  - ◆ FrameMaker (for document composition and storing modular content)
  - ◆ Mif2Go for automated conversion from FrameMaker to HTML, help and other hypertext-based formats)
  - ◆ WebWorks Publisher (for automated conversion from FrameMaker to HTML, help and other hypertext-based formats)
  - ◆ A variety of add-ons and easy-of-use utilities
  - ◆ Acrobat (for conversion to PDF)
- 3 Define a clear-cut set of rules for what equals a “container” of information, what information is allowed in that container, where that container can be used, and use a media-neutral writing style
- 4 *Must* build very rigid templates using the tools you’ve chosen



**with the analysis done...**

The final production flow might look like this



**pros**

You can

- “step” into a long term solution
- reduce your time-to-market and development costs by roughly 20-40%
- create reusable chunks of information that are usable by anyone with access to your system
- create multiple information products from a single, updatable set of source files
- make changes to multiple information products quickly and accurately
- minimize the time to implementation (as compared to a high end system)
- teach others how to use your “manufacturing” system (because one now exists!)
- develop a structure that improves the readability and flexibility of multiple information product
- use off-the-shelf tools

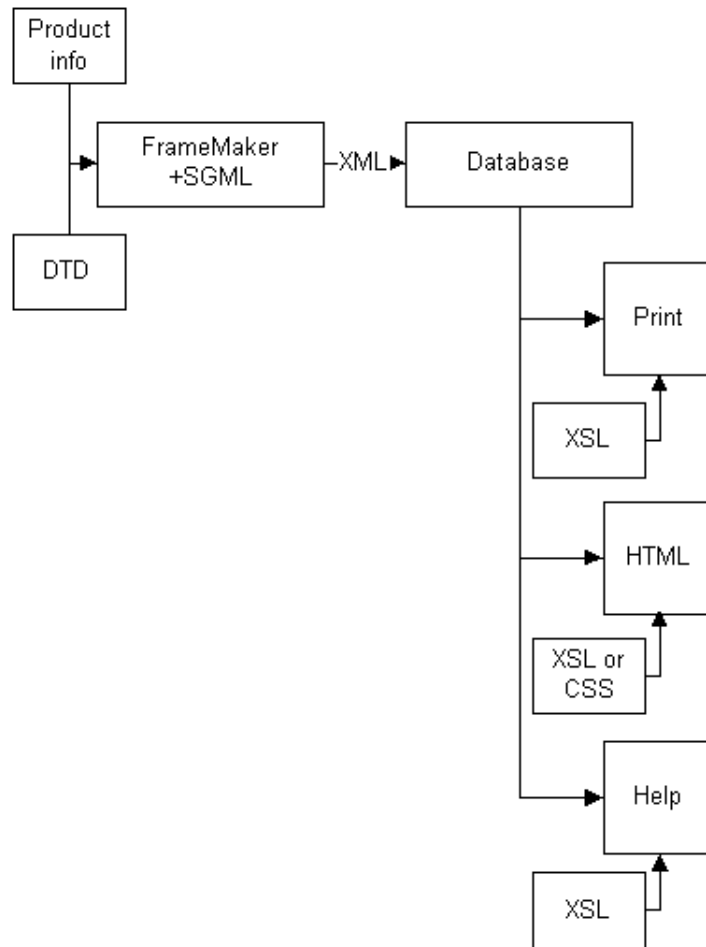
**cons**

- A document composition tool isn’t a database and doesn’t have the same flexibility.
- You must develop a highly structured approach that often frustrates and offends content developers.
- You must plan – everything. Nothing can be left to chance. No “I’ll just do....”.
- You may need to get buy-in from multiple departments (although we’ve had success implementing this approach solely within a department).
- Linear information products (e.g., printed documents) and non-linear information products (e.g., Help files, websites) are used in very different ways, so you either do some heavy compromising or very intense structuring.



## single-sourcing: the high end

Hoo, boy! Here you're into a whole different realm. This is content as data and it's the ultimate in flexibility and reusability. It might look something like this



Here you're working with databases, queries, and mark-up languages like SGML or XML. Why XML or SGML? Because they permit you to define *and* enforce structure, which means do about anything you want to with it. The content stored in your database is just that – content. And that content hasn't a clue how it should be formatted or where it should go. It just knows that it's got characters.

There are no issues with formatting trailing along for the ride and causing a conversion problem, because the formatting, content and structure that we think of as a “document” are completely separated. With that job done, you can then assemble, reassemble and manipulate your content the same way you would work with any other database.



In fact, there are separate techniques for dealing with format, enforcing structure and, in general, making sure that the rules are followed.

---

**pros**

This approach has all of the pros associated with single-sourcing: a middle way, plus

- you have total flexibility
- you can centralize your document development operations into one department
- you can use the same content for multiple departments
- you have the ability to change approaches or write new “rules” as new requirements come along

---

**cons**

- You must develop a highly structured approach that often frustrates and offends content developers.
- You must plan – everything. Nothing can be left to chance. No “I’ll just do....”.
- You will need to get buy-in from multiple departments.
- Linear information products (e.g., printed documents) and non-linear information products (e.g., Help files, websites) are used in very different ways, so you must develop a media-neutral database and writing style.
- You need many new, and potentially different, skillsets to round out your team of experts (e.g., programmers, DTD developers).
- You may not be able to really use any of your legacy information because it must all be rewritten using the new structure and content rules.
- You must provide extensive training, not only on tools such as ArborText, but on how to write for reuse, how to search existing material to see if what you were planning to write is already there, how to write as part of a team instead of as lone ranger.

As you can see, an advanced single source is a formidable job to take on. So why do it? Well...

Imagine being able to generate manuals literally on the fly. A customer logs on to your website, gives you the model he owns, along with other pertinent data, and his preferences for what he wants in his manual. One customer may want reference tables; another may want more procedures. With an almost audible “whoosh” the manual appears on his browser. Don't like the way your documents are looking? Change the XSL. Instant new look. Even the sky isn't the limit with this approach.