Reusing Information: Database Publishing and Other Methods

Yvonne DeGraw

Reusing information in a document set can save time and improve accuracy. This paper discusses the goals and benefits of reusing information. It also discusses techniques for reusing information — symbols, conditional processing, file libraries, and database publishing.

Introduction — Why Reuse Information?

Technical communicators are being asked to produce more documents—both online and printed—in less time. Many of these technical documents contain interdependencies, repeated information, or must be use for several purposes. In addition, many of these documents use information that is also stored in a database or spreadsheet and must be retyped or copied into the document and then formatted.

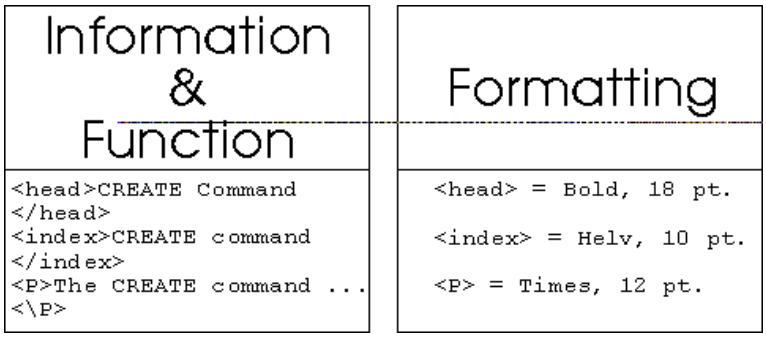
A number of strategies allow you to remove such redundancies in your documents.

Removing redundancy by reusing information allows you to do the following:

- Save time creating new documents—shorten the production cycle. Though planning is required up-front to design a document that reuses information, you will usually save time overall. If you need to create new documents that use the same information, you can create those documents much more quickly.
- Save time (and frustration) maintaining documents. Imagine that your company decides to change the product name in the second version. If you reuse this information, the change is simple.
- Make documents more accurate and consistent. Suppose you store each piece of information (for example, each command name) in a single place and reuse that information. You can change a piece of information without worrying about all the other places it needs to change.
- Be proactive. We have found that reusing information improves our ability to see patterns in the information, to find errors in documentation and software, and to serve the needs of our customers.

Is SGML the Answer?

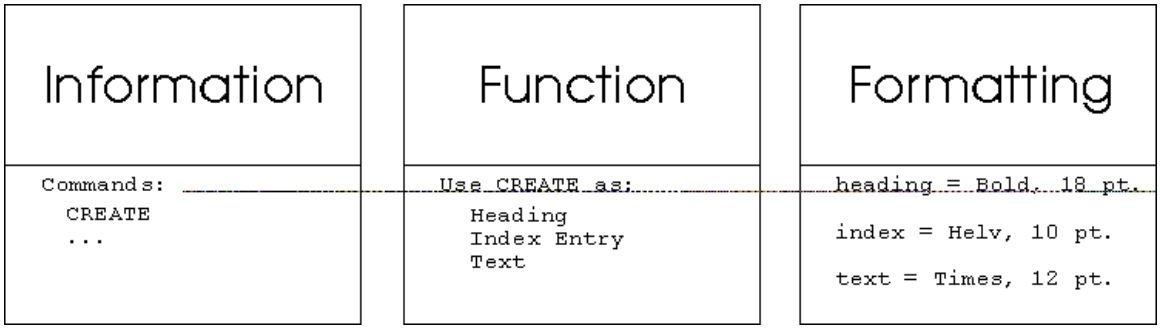
Standard Generalized Markup Language (SGML) is a tag-formatting system that claims to make documents reusable by separating text from formatting instructions. SGML tags indicate how each piece of information is used within a document. For example, a heading might begin with the <SSTITLE> tag and end with the </SSTITLE> tag. A separate file stores the formatting instructions for these and other tags. This file interprets the tags when you create the output.



This type of tag formatting is often used when multiple output formats are required. For example, you may need an 8 ½" x 11" printed document, a 5" x 7" printed document, and online help.

However, tag formatting by itself does not provide ways to reusing information in different orders, combinations, and in different ways. For example, you may use the same piece of information as a heading, as an item in a table, and as an index entry.

To make information truly reusable, we must store text, function in a document, and formatting separately.



Many publishing tools can reuse information in various ways. SGML can be used, but is not the only answer. However, for the more sophisticated reuse techniques, the publishing tool should have some underlying tagged language. For example, FrameMaker has MIF (Maker Interchange Format) and MML (Maker Markup Language). Microsoft Word has RTF (Rich Text Format).

What Information Can You Reuse?

You can reuse any information that is structured in some way. These are some examples of structured information:

- catalogs
- directories
- rate sheets
- schedules
- part lists
- financial statements
- command references
- function or routine libraries
- references for object-oriented software

Some information is more structured than you think. For example, task-oriented procedures follow a structured format, too.

Another example of structured information is software. If you document software, the software may reuse some of the information in your documents. Developers often appreciate ways to avoid hard-coding information into their software, just as we benefit from information reuse. If documents and software share information, you can have much greater confidence in the accuracy of the documents.

The information may already be stored in a format that you can reuse. For example, your company may have databases, spreadsheets, data libraries, graphic libraries, or CASE (Computer-Aided Software Engineering) tools that contain information you want to reuse.

How to Reuse Information

These are some general techniques for reusing information. Different publishing tools use different terminology for these capabilities.

- Using symbols for product termsIf your tool has some way of defining variables, you can use this capability to reuse names that may change. For example, we use symbols for all product names, document titles, and software version numbers. This has come in handy for us because our newest software package has had four names in four releases.
- Separating information into topic filesWhen many people talk about document databases, they are really talking about having a directory that contains a separate file for each major topic. You can mix and match these files as needed. This is particularly handy for products that are sold with various subsets of the complete set of capabilities.
- Storing graphics in libraries.As with text files, you may be able to store graphics in separate files and reuse those graphics. To do this, you will want to find a way to reference the files so that if you change a graphic in the library, all the uses of that graphic in your documents will change. Your software may call this capability "Import by Reference", "Object Linking and Embedding (OLE)", "Publish and Subscribe", or "Live Links".

  Using conditional processing.Tools that support conditional processing allow you to use one set of document files for multiple purposes. For example, one set of files can contain all the information for software that runs on multiple hardware platforms. You can use conditions to print documents for individual platforms or groups of platforms. Another use for conditional processing is to specify which information to include in the printed version of a document and which information to include in the online version.
- Using a database to store informationThe most powerful method of information reuse is to store information in a database of some kind and to create procedures that create documents using this information. The database may be a data file, relational database, spreadsheet, or CASE tool.

  For example, if you are documenting a set of programming functions, your database might contain separate tables to store information about the functions, arguments to the functions, status codes returned by the functions, and relationships between functions. Your procedures could use this database to create an complete reference document, a quick syntax reference, function lists or tables, online documentation, a status code list, related topic cross-references, and more.

  Another example is managing a membership list — for example an STC membership list. You might use the database to print directories in alphabetic order and zip code order, mailing labels, name tags, and various letters.

  There are several ways to integrate such database information with documents:

  → Macros created with some publishing tools can import and format data. You may be able to stretch simple capabilities such as mail merging to accomplish this.

  → A complementary tool (such as Brio Publish for FrameMaker or Ventura Database Publisher for Corel Ventura Publisher) can integrate database information with your document without programming.

  → You can use reporting tools provided with your database, spreadsheet, or CASE tool to create documents in a tag markup language understood by your publishing tool. For example, your database tools could write reports that add SGML tags to the information.

Bibliography

Bergantz, David M. and Diana Margaret. "Object-Oriented Documentation: Something to Sink Your Teeth into," Proceedings of the 39th Annual STC Conference, 1992.

Database Publishing(white paper). San Jose, CA: Frame Technology Corporation, 1991.

Jones, Patricia L. and Kelly M. Doyle. "Modularizing Software Documentation." Proceedings of the 35th Annual STC Conference, 1988.

McClelland, Patricia J. and Alison Boudrel. "Write Once, Use Many: Why and How We Make Product Information Modular. Proceedings of the 40th Annual STC Conference, 1993.

Raven, Mary Elizabeth. "Can Principles of Object-Oriented System Documentation Be Applied to User Documentation?", The Journal of Computer Documentation, Vol. 18, No. 1.

Sametinger, Johannes. "Object-Oriented Documentation," The Journal of Computer Documentation, Vol. 18, No. 1.

Weiss, Edmond H. "Isomorphism Between OOP and Documentation: Reflections on Sametinger's 'Object-Oriented Documentation'," The Journal of Computer Documentation, Vol. 18, No. 2.

Yvonne DeGraw
Sr. Technical Writer
SmartStar Corporation
120 Cremona Dr.
Goleta, CA 93117
(805)685-8000 x312
yvonne@smartstar.com

Yvonne has documented application development and database software for seven years. She has used database publishing and similar information reuse techniques for five years. She was a founder of the Santa Barbara STC chapter, and is currently managing the Santa Barbara chapter's Technical Art Competition.