

Case Study for Single-Sourcing Report Software

By: Joan Thomas

Non-STC member

New article, first submission to a journal.

Summary: This case study outlines how Transcentive, Inc. creates a sample reports document (i.e., sales tool) from the product's user reference source documentation. The article graphically illustrates how our single-sourcing / re-purposing process works and includes an overview of where we began, what we learned along the way, and where we go from here.

Article:

Two years ago, the Information Development team at Transcentive, Inc. (a leading solutions and software provider for stock plan and global entity management) started down the path of "single-sourcing" and "re-purposing" its software documentation.

Our vision was to create source information using Adobe® Framemaker® 6.0 application (<http://www.adobe.com/>), and then re-use it in different ways to increase our productivity while generating quality documentation.

To single-source from those files, we planned to:

- Create books.
- Create online PDFs for distribution via CD-ROM and the company Web site.
- Create software help systems using Quadralay's WebWorks Publisher 7.0 application (<http://www.webworks.com>) for Transcentive's web-based and client / server financial reporting software.

To re-purpose from those files, we planned to:

- Convert file excerpts into other formats (such as Microsoft® Word 2000, <http://www.microsoft.com/>), for other groups in the company—sales, training, development, and support. For example, we used Acrobat and Mif2Go by Omni Systems, Inc. (<http://www.omsys.com/>), one of the many available software plugins.
- Create Sample Report documents. The sales department uses these to present customers with detailed samples of the reports that the Transcentive software generates (www.transcentive.com).
- Create standalone documents such as Release Notes and Product Alerts for support and Transcentive's customer Web site.

To demonstrate how our single-sourcing / re-purposing process works, we present this case study that outlines how we create a sample reports document (i.e., sales tool) from the product's user reference source documentation.

The following discussion presents:

- Where We Began
- Our Process In Action
- What We Learned Along The Way
- Where We Go From Here

WHERE WE BEGAN

Planning, planning, and more planning. This process took two years and continues today during our weekly staff meetings.

In the initial planning sessions, these topics / tasks were addressed:

Buy in to the concept by writers and product management—Could our department afford the investment in time for the planning / implementation, and money for equipment?

Evaluate software product tools—Is Framemaker robust, flexible, and reliable enough? Is support readily accessible? Is the help system conversion process fairly seamless? Does everyone have a comfort level with the software?

Define the writing style—Do we agree that user reference material topics should be written in a functional format rather than menu driven style? Do we agree that we should use a “structured writing” style (Rockley 2003, pp.350–354)? For a description of the basic structures, see Figure 1. Each topic begins with cross references to main topics (Heading 1 level), then conceptual description of the function or process, a numerical procedure, software screen shot, and field descriptions for the dialog box or Web page. These are the building blocks that allow us to re-purpose the documentation.

Create and adhere to templates based on the defined writing style—When files with set styles are converted to a help system or into different sized documents, does the process run smoothly with consistent results? Can we adjust the template? Are there templates for all document types?

Define source and text inset file sizes—We set these files to the user reference size of 7x9 inches, so that template size adjustments are minimized. Sample Reports and standalone documents such as Product Alerts or Release Notes were defined as 8.5 x 11 inches.

Cross-train—Each member of the writing team had a special role to play in communicating problems encountered, special needs, and design issues. Sharing of tips and shortcuts was essential.

Adhere to a file naming convention and directory structure—Whenever possible, source files were saved to the same three folder names in each directory: Frame, Graphics, and Stuff—to maintain cross references and graphic file links. For details, see Figure 2.

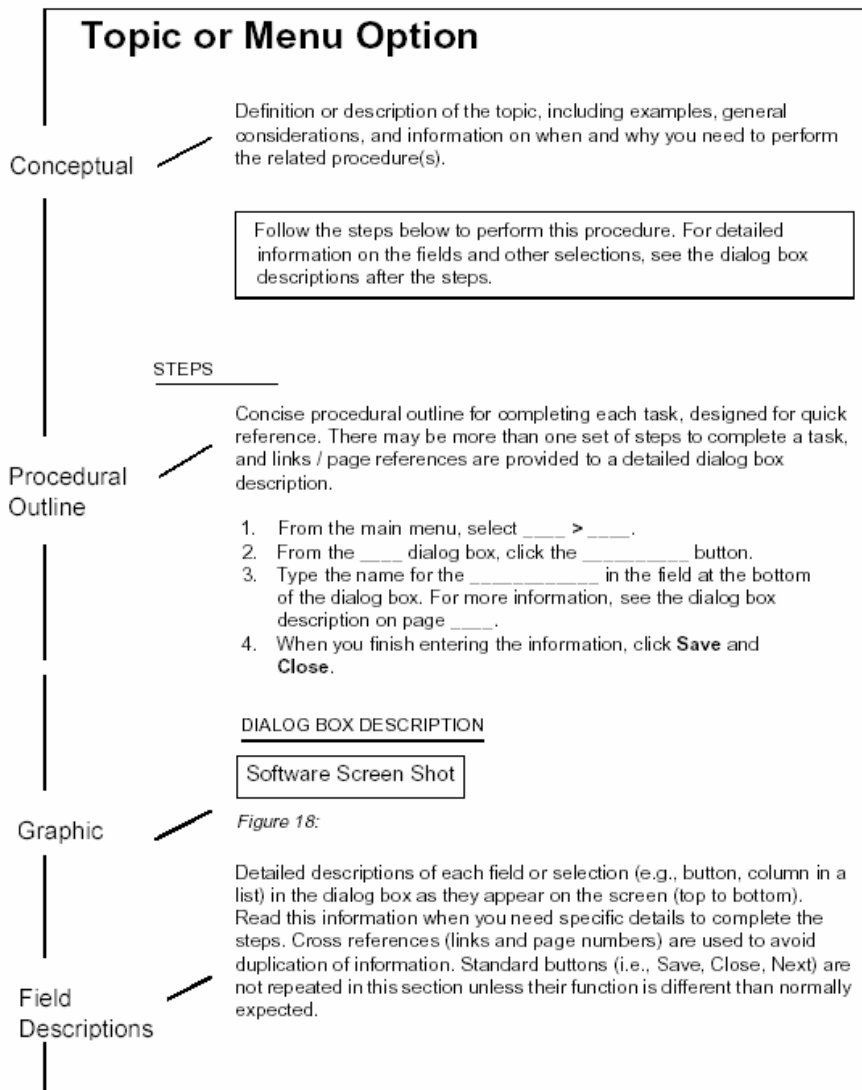
Document the template, tags, and style use—We created a centralized instructional document so that the next member of our team would know why we created the tags we did or why we did it that way (and for the days when we ourselves couldn't recall why).

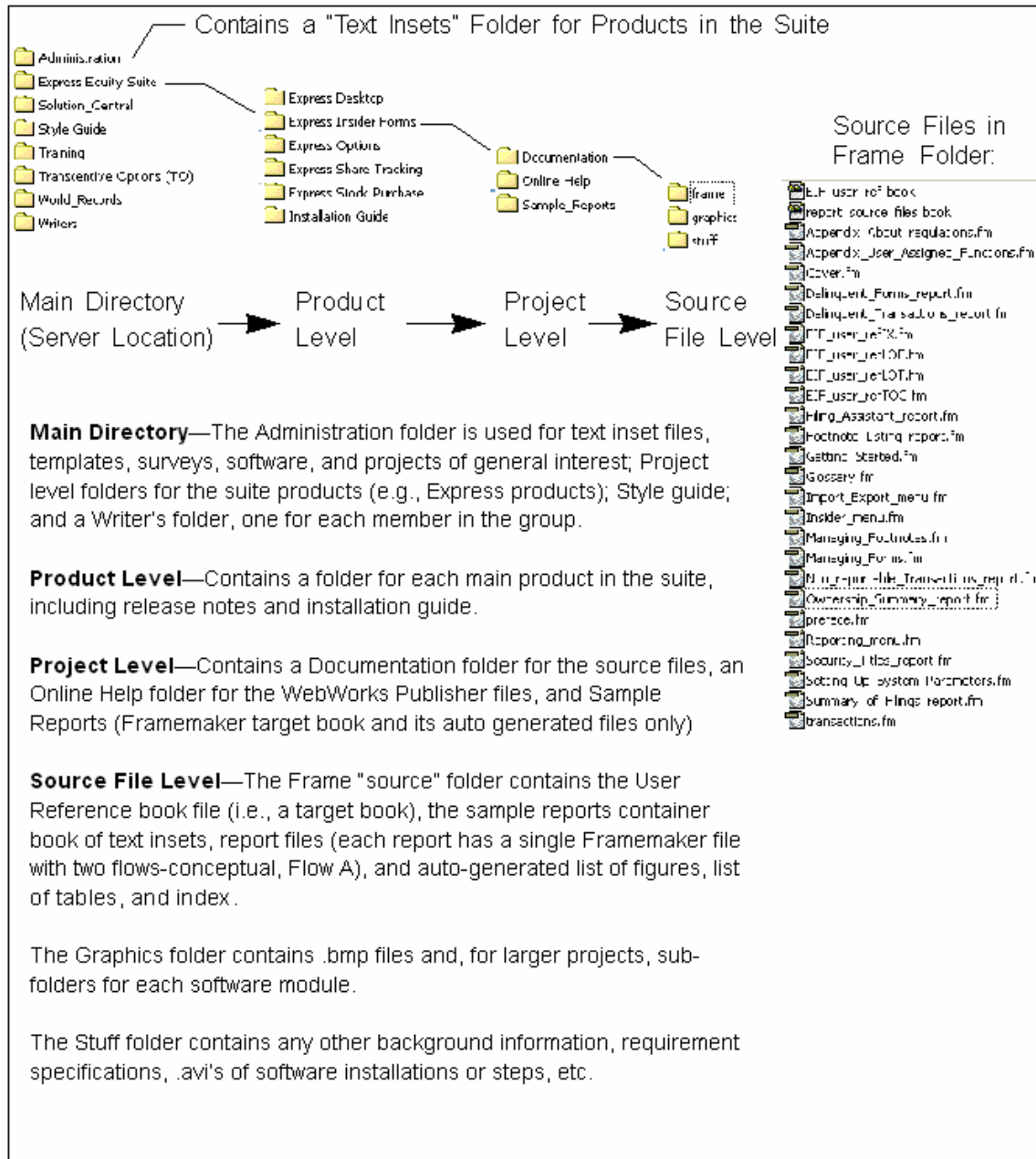
Adjust to a group writing process—There was an adjustment period for writers who historically worked independently, because now the “source” files were common files. Collaborative authoring is often a result of discomfort and resistance (Rockley, Kostor, and Manning 2003, pp. 365–388). You may be the lead on a project for one release, but only work on one aspect of the project next time around. Taking the initiative to communicate with the other writers is an ongoing process. You must learn to say ‘what you did’ causes me to “do this”. How can we both agree? Performing a post-mortem on what worked and didn't work during a release provides a tremendous benefit to all involved.

Today, we continue to evaluate and revise how our documentation is created and handled, always keeping in mind the ripple effect a minor change can have down the line in the process. Our changes are always implemented on current documents (we don't look back too far).

Document Structure

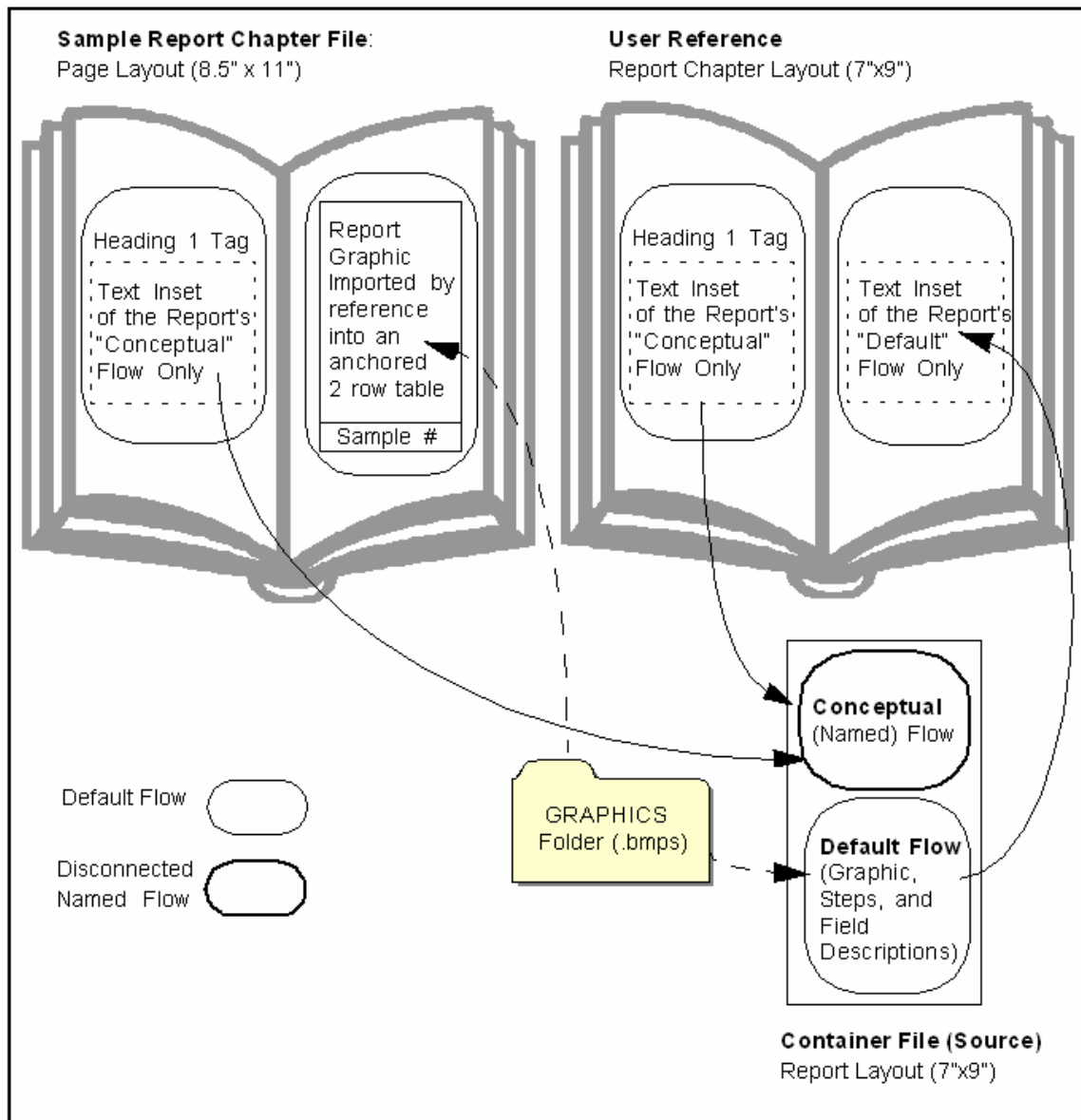
The information presented in Transcentive's User Guides is presented *functionally*, i.e., the way you would normally follow a particular procedure, rather than following the structure of the software. Each topic is organized in the following way:





OUR PROCESS IN ACTION

The Sample Reports document was designed to use Framemaker text inset flows and linked graphics (see Figure 3 for detail).



Notes about our Insets—We refer to re-purposed source files and named text flows as text insets. The book file which contains the inset files we use for a project is called the container book.

The file (to which we import the inset) is called the target file, and the project book (which contains these files) is referred to as the target book.

Our text insets are located in two locations—the Administration folder (for cross project topics such as the preface and legal blocks), and the project's Frame folder (for sample report text inset files).

Sample report insets are located within the source folder to facilitate cross referencing.

Named destination hypertext markers are used when referencing graphics and when cross referencing between files in a book. For simplicity, the use of references is limited to those within text inset files.

We needed to make the following adjustments to the document structure of the report files in the user reference source files to re-purpose the files for a Sample Reports book:

- Created a separate 7x9 inch source file, i.e., text inset source, for each report.

To distinguish these files from other user reference files saved in the source Frame folder, each text inset file was named with an underscore report (e.g., ReportTitle_report.fm). We realized afterwards, that if we had named the files “report_ReportTitle.fm” instead, it would have been easier to sort the file list.

- Formatted each report text inset source file so that it contained two named flows—Conceptual and Flow A (default main flow).

To create this, we inserted an anchored text box and assigned the name “Conceptual” to it. We then pasted in the conceptual portion from the user reference file, and pasted the remainder of the report file into the main text box (Flow A).

- Formatted each report text inset source file so that it ended with the paragraph override. For the last paragraph of each inset, except when the flow ended in a table or anchored graphic, we had to set Run-In Head Default Punctuation on the Pagination tab (Framemaker 6) to eliminate the extra line associated with the inset file.
- Where necessary, the user reference conceptual flow section was re-organized so that the text inset file did not contain more than one page of information in the Sample Reports book target file.
- Cross references were minimized, eliminated, or when necessary, we used hypertext named destination type references.
- Applied conditional text for information that is specific to a single product in a software suite, e.g., show_4_productabbreviation. Never apply two or more conditions to text (multiple conditions usually display in magenta color).

Insert the full name of the inset file as an internal note conditional text. You can now copy and paste the file name into the Window’s Explorer search box to locate unlinked inset files quickly. Likewise, insert a linked graphic’s file name as a File_ID condition to locate or replace the file when needed.

Use the Auto-Text plug-in (<http://www.siliconprairiesoftware.com/>) to create pre-

conditionalized cross references for print only situations, for example, page numbers.

- Created a “container” book file for managing the sample report text inset files. Framemaker’s spell check and search / replace features do not check within text inset files of a book (also the marker management plug-in, IXGen, does not work with text insets, <http://www.fsatools.com/>). Therefore, this container book is used for the purposes of selecting files, spell checking, and searching / replacing information or making style changes.

The book files (i.e., User Reference, Sample Report book) that use the container’s inset files is referred to as the target book.

- After all the report inset files were created, we replaced the user reference source file information with two text insets per report—Conceptual and Flow A.
- Created a Sample Reports book file that contained a chapter file for each category of Sample Report, e.g., Administrative, Tax Reporting, etc.

Within each chapter of the container book, each report consisted of a Heading Title, Conceptual flow inset, and a sample report graphic (linked by reference to the source file).

In the target files, we manually adjusted the main flow to force elements within a text flow to the next page when necessary.

- In the Sample Reports target book files, we used the Tablecleaner plug-in (<http://www.frameexpert.com/plugins/tablecleaner>) after all final reviews were completed to correct format changes that Framemaker does not automatically clear.

Process Summary

The following summarizes the process we used to create the Sample Reports document:

1. Define project deliverables—document size, format outputs, etc.
2. Define inset types and use.
 - Will there be imported named flows, linked graphics, or entire imported files?
 - Can the use of heading titles within it be minimized (e.g., Heading 1 for a user reference, but not applicable in the Sample Reports book)?
 - What naming convention will be used for the inset files and/or named flows?
3. When needed, adjust source files for formatting and named text flows.
 - a. Create a separate “conceptual” named flow for conceptual report information.
 - b. Apply the paragraph run-in head default punctuation to the last paragraph in the inset file to eliminate the extra spacing the inset creates in the target file.
 - c. Minimize, eliminate, or change cross references to named destination hypertext markers. For example, to cross reference between files, use the hypertext format “gotolink filename.fm:linkname”.
 - d. Apply specific conditional text if needed.
4. Create a container book of text inset files for the project. Locate them within the same folder as the project (e.g., Frame) or in a central location for suite projects (e.g., Administration > Text Inset > EESuite).
5. Create the Sample Reports target book, import the conceptual flow, and then anchor a table for the sample report graphic.
6. Replace original material in the user reference source files with the newly created text insets.

One important item to note is that we always have a back-up file set, especially as soon as a project is complete.

WHAT WE LEARNED ALONG THE WAY

In summary:

- Start small with the inset process. Get used to how they work. Use them for preface and legal blocks of information. Work your way up to cross-product information like import / export file layouts and field descriptions.
- Treat imported graphics as a type of inset. You must be aware that because the graphics are shared with different projects, you cannot just modify the source file, for example, change the image size. Always adjust the graphic size within the

target file within Framemaker. Record the file name, conditionalized for example as File_ID, so that you can easily locate files with broken links.

- There is a learning curve. All writers should be familiar with the process, and when feasible, the same writer should be involved with a particular inset project. For example, the same writer who creates the inset files from the user reference files, should be the same one who updates the user reference files with the new insets to see how the process works.
- Manage your insets - remember spell checking, indexing, and search / replace functions skip insets (files and imported flows) in a book. Create a separate container book for these files. Create insets from different named flow tags or use single files.
- Minimize the use of links in the inset files—use named destination hypertext markers when possible. Apply a special character format, such as a blue color to indicate a link only to the link words. When hypertext links are created, typically the marker extends to where Framemaker detects a character tag change. Hypertext Links more powerful for cross file links, but more time consuming than regular cross references.
- When creating a standalone version of a large chapter file with external cross references to other chapters in the book, you can set the hypertext link color in the standalone target file to black to mask the broken hypertext link in PDF.
- There are instances where page numbers within the text inset file do not update properly in the target file. Creating a cross reference marker to the target file in the inset file will create a connection so that the page numbers can update accordingly.
- When using named flows, be aware that sequences, such as Figure and Table numbers, can become disconnected if the items are not all contained in the same flow. Generating a list of figures and list of tables, even when not included in a project, is a good tool for catching errors in sequence numbers.
- Conditional text should be limited to a select few types, for example, internal notes, reviewer notes, print only, help only and project only conditions.
- In some cases, you just cannot create one inset for both uses. For example, the style for writing steps in the user reference is different than in the training guide. In this case, the container document can include both sets of procedures. This makes updating the documents easier in that you can apply the same change to both documents at once to keep them in-sync.
- The text inset process doesn't work for all documents, for example, a course guide. We are currently evaluating the use of Framescript

(<http://www.frameexpert.com/>) to mark areas of text to extract from the source files.

Finally, the Information Development team learned that reporting software offers a unique opportunity for single-sourcing and re-purposing our technical writing—if you plan ahead.

WHERE WE GO FROM HERE

The next single-sourcing project involves creating a training course guide from the user reference materials. This project will be challenging for all writers involved.

We will continue to evaluate our tools, style, templates, usage—what works and what doesn't. After all, nothing remains constant but change itself.

REFERENCES

Rockley, Ann. 2003. "Single-Sourcing: It's About People, Not Just Technology." *Technical Communication* 50, no. 3:350–354.

Rockley, Ann, with Pamela Kostur and Steve Manning. 2003. "Managing Enterprise Content: A Unified Content Strategy." Indianapolis, IN: New Riders.

Biosketch:

Joan Thomas is a Senior Information Developer with Transcentive's InfoDev group, a development team providing documentation support to product teams within the corporation. She contributes to the design and management of Framemaker templates, WebWorks Publisher help systems, and other internal process documents in the company. She is a contributing author to the award winning Express Options Online Help project (2002 Central Connecticut Technical Communication Competition, Award of Excellence). Contact information: jthomas@transcentive.com