An overview of indexing methods

by Jan Wright

You can see the end of this project. Final color corrections are being made on the art, and the appendices are still up in the air, but it's almost done. There's just one little problem remaining - you haven't got the index, and no one seems to be volunteering to write one for you. And your publishing software won't automatically write a good one either - someone of the human variety has to do the entries. What are you going to do?

Panic!

That's a realistic option, and one that's often used in this situation. Professional indexers get late-night phone calls from panic-stricken editors all the time. Authors of trade books often don't realize that their book contract lays the onus for creating and paying for the index on them, not the publisher. Lots of people panic at the thought of writing an index, so you're in good company.

If you are at the panic phase, the time to carefully plan for your index is long past. You can always call a professional indexer to come in and do it for you. But you will probably want to figure out how to plan for indexing next time so that it doesn't drive everyone crazy.

Indexing is a mystery to many people who are writing and printing materials. An index is an offering to your readers - a way in to your material, a subject finder and a detailed guide to the contents of your piece. Indexing itself is a precise art, with not much real mystery when you get into it deeply. It is frightening or annoying because it is very detailed, and requires knowing how to provide access to information as well as the picky details of getting software to deal with topics correctly.

Traditionally, in the Jurassic era before the invention of memory circuits, professional indexers used 3x5 cards to help them build an index. Most have now graduated to using sophisticated stand-alone indexing software, packages that alphabetize immediately, handle typesetting codes, import and export all kinds of files, and provide an unlimited selection of special formatting tools. While the professional indexing world was concentrating on making these software packages the powerhouses that they are now, the desktop publishing revolution was also going on, and publishing itself started moving online. Indexers are caught up now in the same bind that the publishers are: how do you make these DTP tools build quality indexes? How do you fit indexing into a publishing schedule? How do you get the files indexed at the same time you are making final art corrections?

> *"Indexers are caught up now in the same bind that the publishers are: how do you make these dtp tools build quality indexes?"*

Let's look at some of your options for getting an index into your piece. The choices below are listed in order of the amount of effort they take, so you can choose one you have the time to invest in.

Do nothing

Not a good idea. Unless you are publishing fiction or an alphabetically-organized book, you probably need an index. Publishing a book without one tends to make people look at you over the top rims of their glasses. You know the look: former librarians like me perfected it years ago, and it's guaranteed to turn the "Auto-guilt" option on in your Preferences dialog box.

Hire a professional indexer to index in stand-alone software

Stand-alone indexing software allows indexers to write indexes without needing access to your files. The software is specialized for index writing, and includes many features and shortcuts not available in the indexing modules found in programs like Microsoft Word or Adobe PageMaker. It is one of the fastest ways to write an index. If you decide to hire an indexer to write your index in standalone indexing software (not embedding codes), you will need to have

(Continued on Page 5)

several things in order to help him or her do the work most effectively.

- Having firm page breaks and page numbers helps the process go most efficiently.
- Have the book's text finalized. If you have the indexer work from a draft, the indexing will have to be checked against the final text, a time-consuming process.
- The indexer will need to know what style index you want produced, whether run-in or nested. Here are examples of the two types:

Nested (indented)	**Run-in**
Barbie Midge and, 21 relationship with Ken, 23-26 relationship with Skipper, 34	Barbie; Midge and, 21; relationship with Ken, 23-26; relationship with Skipper, 34

If you have a writing style guide, pass it along. It will help with style decisions for capitalizing command names or handling special terminology.

- The time frame for indexing and the final deadline for the index are important to know so that the work can be planned. A fast indexer can work through 50 to 100 pages per day, depending on the depth of the material. An additional amount of time must be added for the editing process, which comprises nearly 1/3 of the total time needed for indexing.

  Why does the edit take so long? During the entry-writing phase, the indexer often includes more detail than necessary in each entry. This extra detail is needed during the editing phase. When all the entries are viewed together for the entire piece, the excess detail is eliminated where needed, additional detail is added where needed, and the index is evaluated to make sure that its coverage and depth match the presentation of information in the book. Editing can take almost as much time as writing the initial entries.

  Indexers are sometimes asked if they can provide the index chapter by chapter, especially for pieces that will be translated. This is usually wasted effort. A chapter by chapter approach would give you essentially an unfinished and unedited index. The entries can't be compared against ones in later chapters, and the terminology can't be refined and matched.
- It's easiest for an indexer to work with printouts, so that the monitor screen on the computer can be completely devoted to the indexing software, not reading in one half and indexing in the other. You do not need to give the indexer files, although you can. You can also chunk the book, as long as you do it in order, and allow enough time after the final chunk for the edit.
- You will need to update the indexer with any changes to page breaks, terminology, and schedule slips.

At the end of the process, the indexer can give you a file in a specified word-processor format or in RTF. You would then review it, and either return editorial comments to the indexer to incorporate, or incorporate them yourself. If you are planning to update the book, have the indexer incorporate the changes. That way, the index database will have the most recent data when you do the update. Let the indexer know you will want to keep that data and have him or her do the update.

The advantages of standalone indexing are in the power available in the specialized applications indexers use. Speed, cross-reference checking, sorting by groups of entries, sophisticated search-and-replace, powerful and instant compilation, total control over styling, and a choice of viewing the growing index any way you want mean that writing and editing are easy and the tool makes it very fast. The downside is that you must rely on the indexer to get the page numbers right, and if you change the numbers, they must be updated in the index as well. The software makes it easy to do, but it still has to be done.

Use an automated indexer to generate a concordance

Automated programs such as Sonar Bookends will allow you to build a concordance for your piece. What's the difference between a concordance and a real index? Basically, concordance-building programs use an algorithm to extract key terms from the text, and arrange them alphabetically. Usually the

(Continued on Page 6)

programs have a "stop list" of words that are filtered out and excluded. The text file generated will contain most of the terms and nouns present in your files, with page numbers attached.

To use one of these programs, you should have your page breaks set. Then you choose from some depth and inclusion options, run the program on the files, and place and edit the resulting text file.

The advantages of these kinds of programs is that they require very little thought, and you can have something that looks like an index for your piece very quickly. The disadvantages are immense:

- Editing the resulting text file to get rid of meaningless entries can be more work than indexing it correctly. You might want to run a sample through the program first, and then study it for meaningless terms, such as "Nevertheless, 23, 35, 48" or "Hereafter, 100." All of these terms should be added to the stop list.
- Synonyms for important concepts will not be included, nor will there be any cross references. The terms included in the file can seem haphazard, and the handling of personal names can be very interesting, to say the least.
- The index is not embedded. If you change pagination or change your text, you will need to recompile the index to reflect the changes, and re-edit it.
- Important concepts will not be analyzed for terms other than those appearing in the text. In other words, there are no "other words." If you refer consistently to canines, "dogs" may not appear anywhere in the index. It's also unlikely that "canines" or "dogs" will appear as subheads under "mammals." Related information in several locations will not be pulled together conceptually. The hierarchy of information will not be built. Forget about having cross references.

  For example, if you were writing a manual on a software package that utilizes palettes, don't be surprised if you get an entry from automated software that resembles this:

  Palette, 3, 5-6, 7, 8, 9-10, 11, 12-14, 15, 16, 17, 18,...

You get the idea. Automated index builders cannot distinguish between passing mentions of a term and important discussions of a term. Entries like these aren't useful to anyone. If you expect the book to be really used, this is really not a good option. You are not doing your readers a service, and probably you will get some of those looks I mentioned earlier from people who really care about indexes.

"Automated index builders cannot distinguish between passing mentions of a term and important discussions of a term."

Embed indexing into your publication's files

If you decide to embed indexing using your DTP software, you will need to get familiar with its unique embedding techniques. Writing an index using embedded codes is both easier and harder than writing in standalone software. Automatic page numbers alleviate the burden of getting every page number perfect, and make updating after page break changes easy. But most DTP software packages are closed systems: the indexer must work in the same files that the production people need to be tweaking for final edits and printing. This can put a time strain and file hand-off strain on the indexer that doesn't happen in standalone indexing.

In general, you need to make sure that the machine being used for indexing matches the configuration of the production machines. The software version should be the same, special macros, templates, or add-ons that affect layout should be installed, and most critically, the printer drivers and the font setup should match. Otherwise, font substitution could make the pages change enough to throw off indexing.

Make sure the indexer has a printed copy of the piece so he or she can see the page layout during indexing - most embedding software packages hog the screen when displaying the index modules, and it's hard to figure out where on the page you are, and where the page ranges should end.

(Continued on Page 7)

The text on the pages will float as it is edited or rearranged after your indexing is done, and paragraphs may move to new pages. In traditional indexing, you only worry about page ranging if the text goes for more than one page. In embedded indexing, you should worry about what will happen if the text floats to another page, even though it is all contained on one page now.

The down side of this is if your text is heavily edited after indexing, this page ranging can trip you up. New paragraphs with different styles could affect page ranging and entries will need to be checked in the material before the additional paragraphs.

The indexer basically follows the same process of making entries as in a standalone product, but there are a few additional concerns:

- To do a complete edit, compile and print out the index. Work through the printout making notes on what to change. Then the editing process splits into two phases: edits you can make to existing entries (corrections, spelling changes, deleting) and additional new entries you will need to create in the files. You have to go back to the actual page to add new entries.
- Allow enough time. Indexing well by embedding is slower than in a standalone application. The page-ranging choices are more complex, and at times require counting or bookmarking paragraphs. Each software package places its own demands on your machine's memory, so use a fast machine. The 50 pages per day goal will probably not be reached for a complex piece. Editing will still require 1/3 of the total time.

  Also make sure there are large chunks of time in your schedule for the indexer to have access to all the files. Trying to break editing work into file-specific pieces is very hard with an index. It wastes the indexer's time if he or she has to wait for files. During the editing process, nearly every entry gets changed somewhat, and since editing happens in alphabetical order, that means edits happen one-in-this-file, next-in-that-file, back-to-this-file, two-in-section-three, haphazard order.
- If anyone works with the files after indexing, make sure they know not to delete the index tags. Put a process in place for rewriting that establishes how to move the tags when a paragraph moves, and how to handle deleting text.
- Put a process in place to let the indexer know about any changes in terminology, page breaks, etc.
- If the text is heavily updated, allow time for the indexing to be checked on those pages. Checking indexing for a page requires checking the index tags to see if they still apply. This is very time-consuming! It's one of the reasons you should wait until nearly the end to index a piece.
- Many of the programs do not handle small caps or italic text you need to have in the entries. Adding a *** to the end of problem entries will allow you to search and replace for special text in the index text file at the end.

  At the end of the process, the index is compiled and placed into your publication. Keep in mind that this index is a snapshot: if you move or change text in the files, the index is not automatically updated. You must recompile to get those changes into the index file.

"Writing an index using embedded codes is both easier and harder than writing in stand-alone software."

Embed a standalone index into publication files after compiling

This is one way around the difficulties of handing off files, or writing and editing using embedded codes. Essentially you have the index built traditionally in a standalone indexing program as described above. After the editing phase, you resort it into page number order, and enter the entries into the appropriate places in the publication files.

Why would you want to do this? It is one way to keep the files free for production changes up to the last minute. If you cannot afford the

(Continued on Page 8)

time it would take to index in the files, but you still want embedded entries, this could be a viable choice.

One beauty of this method is that if you break your pages, and paginate them, you can go to press with the standalone index file, and embed the codes after the book goes to press. This is often a great idea if the book is to be translated or converted to an online format. You get the best of both worlds - speed and codes.

According to Ed Malick of Frame Technology Corporation, this is the technique used for building Frame's manual indexes at one time[†]. Frame went in the direction of embedding a standalone index into FrameMaker files because it allows indexers to develop indexes using the tools that work best for indexing, and keeps them from worrying about final pagination changes. The time it takes for embedding is less than it would be to write the entire index within FrameMaker itself.

To adapt this technique to your software package is easy: provide the indexer with printouts, and have the index written in standalone format. Follow the same requirements listed above for standalone indexing about firm page breaks and changes to the files.

After editing, have the indexer generate a file that shows each entry sorted by page number rather than alphabetically. On a production machine, start the embedding and work your way through the list. Generate an index periodically and check it against the original to make sure it is coming out correctly.

Or if you want, wait until the piece goes to press and then embed the codes.

Not happy?

Not satisfied with any of these methods? I'm not either. Some day I would like to describe to an engineer how my ideal software package would work, in an open desktop-publishing system.

What we need to have to make indexing work effectively is a combination of tools:

- DTP tools for laying out the piece
- Inventory tools for assigning invisible unique codes to each part of the piece
- Indexing tools to index outside of the files to the unique codes
- Options for setting up the index to be print, online, or both
- Embedding tools to strip the indexing into the files when it is ready
- Translation tools to make that function easier

Any engineers intrigued out there?

[†] "Indexing at Frame Technology Corporation," *The Changing Landscapes of Indexing*: Proceedings of the 26th Annual Meeting of the American Society of Indexers, San Diego, California, May 13-14th, 1994. Published by the American Society of Indexers.

Announcements

rticles Wanted: The newsletter is looking for good articles on indexing to help our members learn more about specific techniques, access method theory, software packages with indexing modules, etc. If you have an article idea, please contact Pilar Wyman, pilarw@aol.com, or Jan Wright, jancw@mindspring.com, for more information.

Contact the editors for submission requirements and file formats. The next deadline for articles is April 1, 1998.

uestions for Q & A Wanted: If you have a technical indexing problem that you would like answered, consider submitting it for our Q & A section! We will find an experienced indexer to publish an answer, helping others who may run into the same problem. Send your questions to Pilar Wyman, pilarw@aol.com, or Jan Wright, jancw@mindspring.com.

Watch for our upcoming Web site and listserv!
