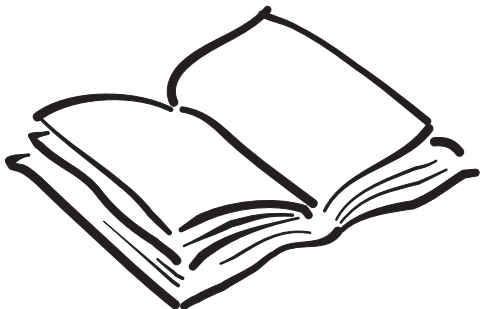

Working with Windows Help Keywords



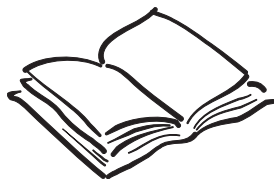
Jan C. Wright

Wright Information Indexing Services

Copyright 1996 Wright Information Indexing Services

Portions of this text previous appeared in the WinHelp Journal, Winter 1996 and Spring/Summer 1996 issues. The WinHelp Journal is available from WinWriters at (206) 285-2605.

Jan C. Wright is the owner of Wright Information Indexing Services, which has provided print and online indexing to companies such as Microsoft, Asymetrix, Visio, PeachPit Press, and Sakson and Taylor. She has served as a judge for the Society for Technical Communication's annual publications competition, has a Master's Degree in Library Science, and uses her many years of experience as a professional librarian to help make her indexes more user-friendly. She is a member of the American Society of Indexers, and recently co-developed the Society's World Wide Web home pages. She can be reached at JANCW@mindspring.com, or at 206-784-2895.



*Jan C. Wright
Wright Information Indexing Services*

*206-784-2895
JANCW@mindspring.com*

<http://www.mindspring.com/~jancw>

Working with Windows Help Keywords

Have you ever been frustrated trying to find a Help topic using keywords? Your users find poorly thought-out keywords just as frustrating to use as you do, especially when each set a person uses is different in scope, in approach, or in the results they get. As more and more information is converted to electronic documents, unpredictable keyword indexes make users extremely frustrated. The range of keywording quality in Windows Help reflects the fact that no standards have been developed. Standards addressing what you should include, how to phrase it, and punctuation standards to set are necessary to develop before you begin indexing your Help systems. Let's take a look at some standards you can set for keywords, and then follow up with ideas for fixing common keyword problems.

Why set standards?

For a user, an online help file is very different than a book. You can't bend the pages in online help, you can't use your favorite old bill stub as a bookmark, and you have no real clue how much information is available. In a book, you remember where you saw a piece of data, its location on the page (perhaps a third of the way through on the left side). There is a linearity to its structure, as well as a physical size and reference. Online, there are no clues except for the table of contents and the keywords. For this reason, your keywords are an extremely important retrieval tool.

Are keywords important when you have full-text search available? Yes, for one simple reason. Readers run out of terminology after a few tries, and like browsing for an answer. If they don't know what to call a feature, they don't know what to type into a search field. For this reason, keywords also help users learn a product's terminology.

Windows 95 changes the visual presentation of keywords, but does not make generating them any easier. Keywords now appear in a book-like index, and its new appearance introduces some new design issues into keyword development. A poorly-built index is immediately visible in Windows 95, because the indenting structure reveals keyword inconsistencies.

Some definitions

Before we talk about what standards you should set for keywords, let's define some basic terms. Let's assume you are using Microsoft Word to build your basic Help files, as that is the standard tool for development.

You embed keywords in a help topic using K footnotes, and if you are using one of the common help development tools, K footnotes will be created automatically for you, often with the topic title inserted as a keyword. Figure 1 shows the location of a typical K footnote marker, embedded in the topic title.

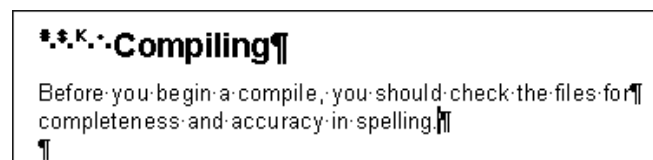


Figure 1: A K footnote marker embedded before the topic title

To view the actual keyword entries for this topic, you must display the footnotes (notes) pane by choosing the Footnotes command from the View menu. Each of the symbols for the footnotes gives you important information, such as the topic ID and title.

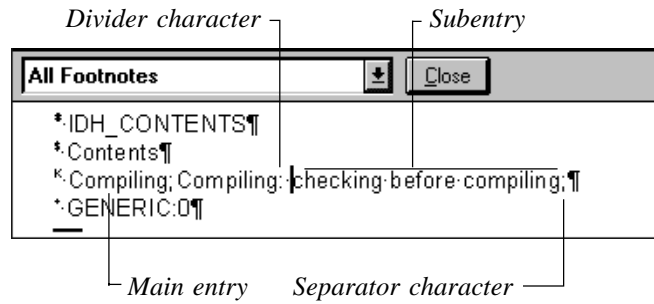


Figure 2: K footnotes showing sample keywords

In Figure 2, the K footnote contains two keywords associated with the topic in Figure 1. This is how you keyword a topic, by typing the keywords into this footnote. You can embed multiple keywords in a topic by using a separator character, most often a semicolon.

“Main entries” and “subentries” are terms borrowed from the world of print indexing, and they mean slightly different things in the world of keywords. Keywords come in two main forms: main entries alone, such as “Help” or “Help button” below in Figure 3, and main entries with subentries, such as “context-sensitive” under “Help.” I refer to the latter type as subentries.

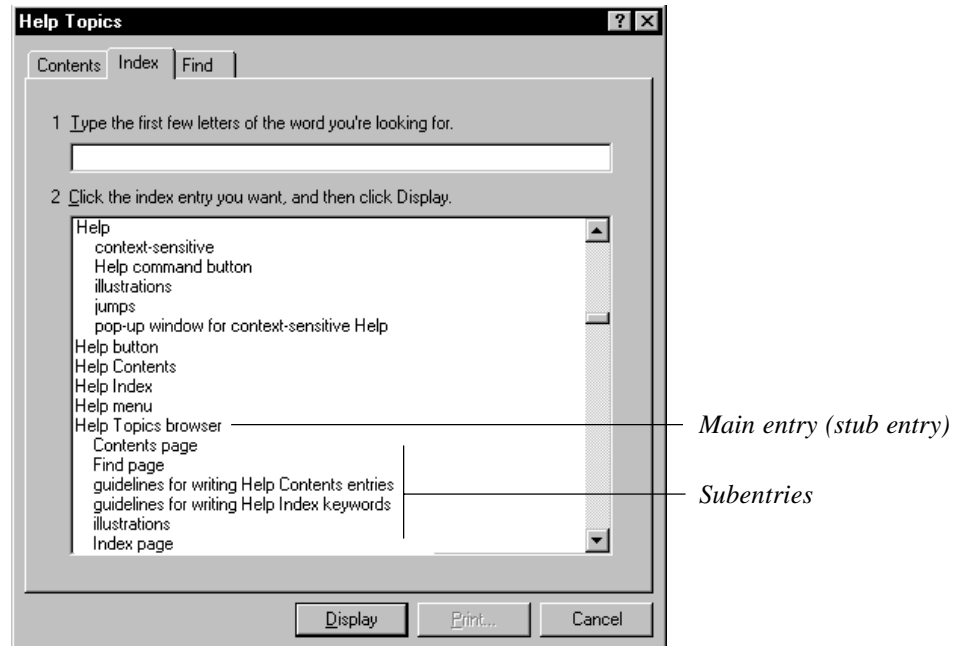


Figure 3: Main entries and subentries

Some writers are using the term “Stub entry” for the main entry that sits alone above the subentries. The main entry text is not repeated in each entry in Windows 95, but the keywords still require it to work. If you refer to Figure 2, you can see how the subentry contains both the main and subentry text.

You must specify a divider character that divides the main entry from the subentry text (a colon in Figure 2). You will not see this divider character or the separator character in the compiled index. For example, in Figure 3, the full set of keywords needed to create the first two entries would appear like this in the K footnote:

Example:

Help:context-sensitive;
Help:Help command button;

You should repeat the word “Help” even though it shows up only once as the main entry (the stub entry) in the index.

Now that it’s clear where keywords live, let’s talk about how to make them more usable.

Avoiding complaints

Developing some standards for your keywords, and taking time to edit them for usability will make your index more predictable and your users less frustrated. Your users will know what to look for, and spend less time searching through long lists of repetitive keywords. If you don’t set some standards, you risk giving your users confusing results.

In Figure 4, notice how the subentries for “Help” are separated and lost, divided by “Help button” and “Help contents” and all their friends. The user has to go through the entire list, plain and with colons, to make sure all the material on Help has been found. They don’t know what the rules are: why are the Help entries at the bottom different from the single Help entry at the top. What do the colons mean? Are all the entries under “Help: context-sensitive” about context-sensitive help, or just about Help in general? No wonder people get confused and complain loudly!

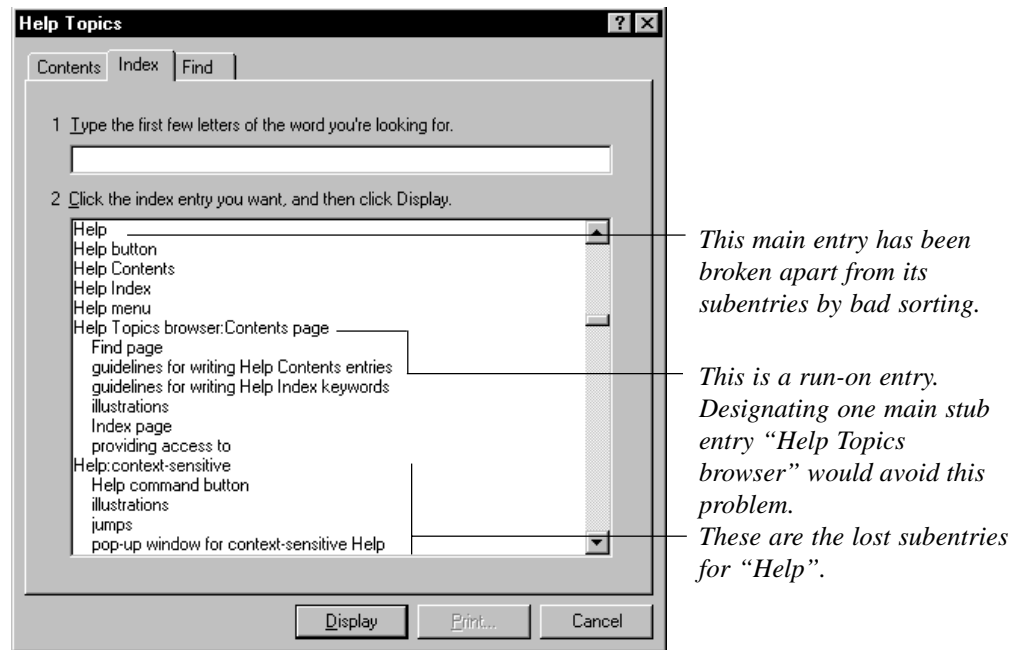


Figure 4.: A confusing section of keywords

Goals for keywords

The goals you should be striving for in keywords are actually similar to those for print indexes. They include:

- ◆ Outlining the terminology used in the product: keywords can provide a guide to what you call things, which users really appreciate;
- ◆ Including the terminology that users already know: keywords can point users to information using their own words and reference points;
- ◆ Insuring that information is easy and quick to locate: synonyms and multiple keyword listings for each topic insures a better first-try hit rate;
- ◆ Providing pointers to information from old or unused terms: updated products often change their terminology, and including older-term keywords can help the upgrading user find the new information;
- ◆ Revealing the structure of the information and its inherent relationships: a well-planned keyword list will show how different topics are related; for instance, color print settings and color monitor settings will show up close together, or all the species of mammals being discussed will be pulled together in a meaningful hierarchy;
- ◆ Avoiding sending users to inconsequential information: weeding out keywords for repetitive topics means that users get to what they need faster;
- ◆ Pulling together scattered concepts into one location: if information is found in many topics on a particular subject, the keywords can be the path into all of them. (Lathrop, 1996, with mutations)

What should you keyword?

Here's a basic product-related list of items that you should keyword. You will need to add to this list if your keywords are for a Help product dealing with specialized subject matter.

Main product features	Field names
Main tasks	Functions, commands, and parameters
Main topics or chunks of information	Data types, methods, objects, or
Menu commands, dialog boxes, and	properties
tab names	Screen names
Important concepts	Synonyms
Acronyms	Troubleshooting information
Definitions or new terms	Warnings, error messages, or restrictions

(Lathrop, 1996, with mutations)

Writing entries and subentries

Help authoring tools such as Doc-To-Help often automatically include the topic title as a keyword, but you should rewrite these titles to be more helpful to users. Listing several topics under "Using" or "About" is not really keywording. So how do you actually write entries? Sit down, pull up the first topic, read it, decide what terms and synonyms really need to be keyworded, open the footnotes pane and start in - there's unfortunately no automated way to do it well. It still takes a human to analyze a topic's importance to a user.

Standardize your style

Before beginning, make a couple of decisions about how you want your keywords to appear, and set up a standard style. Capitalization and punctuation should be absolutely consistent. Once you start keywording, do not switch between initial caps, all caps, or no caps in your keywords. I usually recommend using an initial cap on main entries, and lowercase on subentries; you can use whatever you want as long as you are consistent. The initial cap, lowercase combo is very helpful in debugging later on. Reading all caps is extremely hard on a reader, and should be avoided. Product features that are capitalized in the topics should be capitalized in the keywords.

Example:

Tab stops
 deleting
 inserting
Tab dialog box settings

If you decide to use commas in your subentries, use a different character such as a colon as your separator character.

Punctuation problems

Do not use periods for file extension keywords, as the periods sort above the “A” section. Just use the file extension itself as an entry (BMP files, PCX files, etc.). Similarly, quotation marks will force an entry up to the top. Try to eliminate quotation marks around product features so that they sort alphabetically instead.

If you have automatic correction turned on in Word, you will probably want to turn it off while working with keywords. You do not want any automatic corrections to spacing and punctuation marks happening while you aren’t looking.

We may see sorting problems go away if special sorting controls are implemented in the HTML version of Help. More on that later.

Product vocabulary

Set some standards for controlling your product’s vocabulary: how you refer to product commands and features shouldn’t vary. This part of your standards you can steal from the writers’ style guide, if you have one. You should avoid creating keywords that are one or two characters off, forcing the user to search highly similar terms instead of one standardized term. For example, the keywords below would wind up close to each other, but it would be better to choose one way of referring to features and have all the related topics come up under just one of them.

Example:

set width option
Set Width option
Set-width option

Essential words

Put your most essential words first in your keywords, and use nouns as much as possible. Avoid putting words like “Using,” “The,” “About,” or “Getting” as your first word. No one will look under “U” for “Using the Print command”. They will look under “P”. This is the main reason to edit the automatically-inserted topic title keywords; make sure the most important words come first.

Nouns and gerunds

Make your noun keywords plural, and avoid strange constructs like “Page(s).” If you do use verbs, use gerunds and use them sparingly! Avoid gerunds like “Adding” or “Creating,” as you will end up building a list of everything a product does under those terms.

Do not begin entries with “A,” “An,” or “The,” or with conjunctions like “And,” or “Or.” Use these only in subentries, and only if absolutely necessary for meaning. Try to put them at the end of the entry so that you sort on the most important keywords.

When you have proper nouns, statements, commands, fields, or parameters, present them exactly as they are in the text. You can add qualifying terms to names of features so that they aren’t confusing.

Example:

Parameter numeric field

Parameter text field

When you have chunks of information that are repeated throughout several topics, only index one of them. Adding keywords for repeated and essentially the same information in multiple sites doesn’t help the user.

Creating multiple access points

Add keyword entries that are inverted, so that you have multiple access points.

Example:

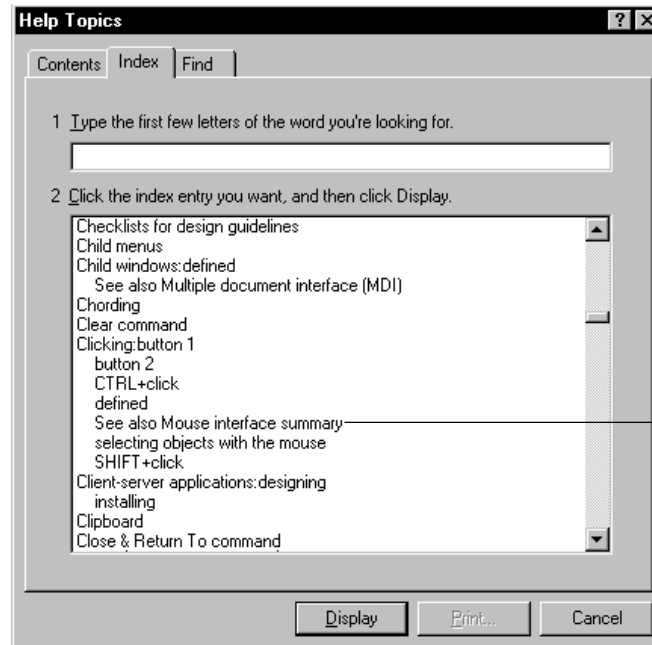
Mail: sending

Sending mail

Use synonyms for your topic, and then assign those synonym keywords consistently for each applicable topic. You should include synonyms that readers are likely to think of, even if that synonym doesn’t appear in the text. Create keywords for concepts as well. You might want to add a keyword for “E-mail” to the example above, even if it does not appear in the text. If you do, make sure that every topic about Mail includes the complete set of keywords: E-mail, Mail: sending, and Sending mail.

Cross-references

Use cross-references sparingly, and set standards for their use. Notice in Figure 5 how the first “See also” reference doesn’t work because it contains subentry text in a run-on before it, and how the second “See also” gets buried because it is sorted it under the S instead of placing it first on the list.



This “See also” is hard to find, and should be at the top of these subentries

Figure 5: Cross references

Use “See” references to lead the user from non-used synonyms to useful terminology. It will probably take a few topics on similar subjects before you think to use a cross-reference instead of repeating all those keywords for each of your topics. Adding cross-references requires some decision-making. Duplicating entries under synonyms is your best choice, but when you have space limitations or there is just too much material to duplicate, or too many synonyms to duplicate it under, use a cross-reference.

When you create a “See” reference, place it as one long main entry, and standardize the spacing and punctuation you use for your references.. In the example below, there are two spaces before the “See” to make it stand out, and the “See” is capitalized. Repeat this style on each “See” reference for consistency. When you click a “See” reference, you will still get topics, so embed this keyword in a topic that has a general overview of the seen topic. For instance, the following reference should bring up a general topic about 2-D shapes.

Example:

Two-dimensional shapes See 2-D shapes

When you want to let the user know that there is more information about a topic under a different keyword, use a variant of a “See also” reference. To add to the example above, a user could possibly find further information about 2-D shapes by also looking under “Shapes.”

Example:

Two-dimensional shapes
Also search for 2-D shapes

“See also” references assist a user in jumping up the hierarchy of information to broader or more general topics.

“See also” references should be keyworded as subentries. Your main entry for a topic should lead to the overviews and general topics, and the “See also” references should lurk right below the main entry. But as we noted earlier in Figure 5, “See also” winds up being sorted under “S.” Try using “Also search for...” as your “See also” reference text, which should pull it up to the top of the subentry list. If “Also” is not going to work, and subentries still come before it, it’s time to get creative and rewrite the problem entries so that they follow “Also” but still have their most important words first.

Another idea is to put your “See also” reference in parentheses. This works out especially well if your keywords are going to be translated into another languages, as the parentheses will sort above all the alphabetic listings no matter what “See also” translates into.

These sorting problems will hopefully go away in the HTML-based version of Help. “See” and “See also” references are also supposed to become active, leaping the user to the section of the index they need to see, rather than displaying topics.

Editing the compiled list

Now that you have worked through all your topics and added keywords, you are halfway done. Yes, it’s true, only halfway. Much of the work that makes a good keyword index comes during the editing phase. This can’t be emphasized enough, as too many products are being shipped with serious keyword problems, resulting in users frustrated with unpredictable indexes. To guard against this problem, you need to see all the entries in alphabetical context to make sure they are consistent, to refine them, and to eliminate problems in sorting. Then you need to fix the problems.

Problems to watch for

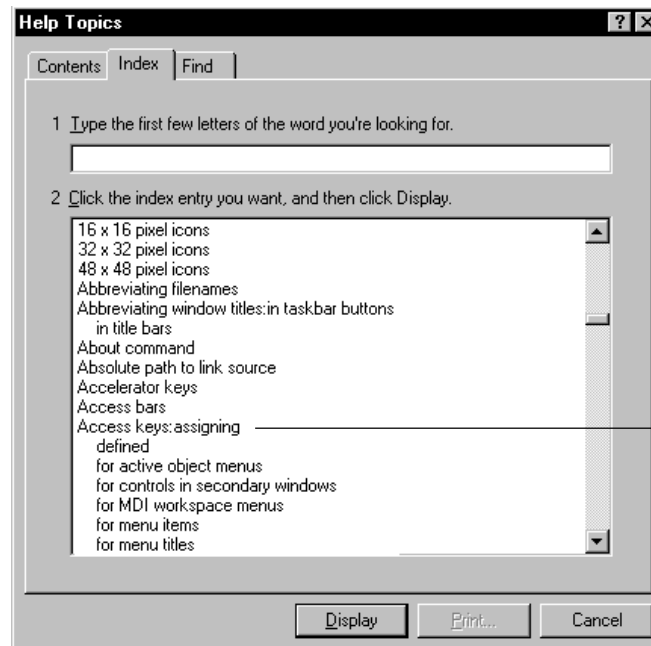
What repairs should you be making? Here’s a brief outline:

- ◆ Eliminate a subentry if it is the only one under a main entry, and points to the same topic as the main entry.
- ◆ The number of keywords should reflect the amount of material in the Help file. For example, if you have two topics on printing, you should not have a set of “Printing” keywords with 15 subentries under it. Likewise, if you have 15 topics on printing, you should have more than two subentries under “Printing.”
- ◆ Consolidate unnecessary subentries down into a main entry. If a main entry has five or fewer subentries and five or fewer topics being referred to, its subentries may be unnecessary. This “Rule of Five” comes from standard indexing practices as well as many usability studies; five is about the maximum that the human brain can remember or distinguish.
- ◆ Examine your main entries (stub entries), which are critical to the Windows 95 indenting structure. Main entries should point to five or fewer topics. If you find a main entry with more than five topics, consider creating subentries for it to help the user distinguish which one is needed. Moreover, main entries that have subentries should be pointing only to overview or general topics. Let the subentries do the work of dividing up the topic.

- ◆ Examine your subentries. Rework long lists of subentries to make them simpler. Remember that readers have short attention spans online, and thus have a hard time wading through miles of subentries. If you cannot avoid the long lists due to the amount of material and detail, don't feel unnecessarily guilty about it. At times, it can't be helped.
- ◆ Check your synonyms for topic consistency. If you have five topics coming up for "Mail: sending", "Sending mail" should have all those topics as well.

Run-on entries

One common problem found in keywords is the lack of a solitary main entry, the stub entry. When you have nothing but subentries and no stub designated somewhere, you will get a run-on head, as shown in Figure 6.



Designate at least one stub main entry to avoid this type of run-on entry.

Figure 6: Run-on entries

Users find this run-on listing very confusing: are all the entries under "Access keys: assigning" about assigning Access keys, or are they actually separate topics about other factors affecting Access keys?

The way to fix this is to always designate one main entry keyword, the stub entry, usually in an overview or general topic. If you wind up without a usable overview or general topic, assign the main entry keyword to a topic that makes sense in that primary position. If you don't have a topic that will work, go ahead and assign the main entry keyword to all the topics. For example, if you have a "Bold text" keyword that brings up three topics on formatting text, assign the keyword to all three topics (less-than-five rule, remember?). However, if you have a "Bold text" keyword that brings up 14 topics on formatting text, assign the main entry keyword to the most important or most general of them.

A related problem with run-on heads is illustrated in Figure 7. This problem can occur when commas in the subentries have thrown off the index. Notice how every main entry that follows a comma-containing subentry has bumped back out and created a run-on. All of these entries are about “Accessibility”, and should be in one long flow of subentries. Most likely a comma was set up as either the divider or separator character for compiling, and commas were also used in the subentries for distinguish meaning. If you use commas in your subentry text, use a different character for your separators and dividers. Avoiding them entirely is actually the best solution, as they sometimes cause problems even with the compile options set correctly.

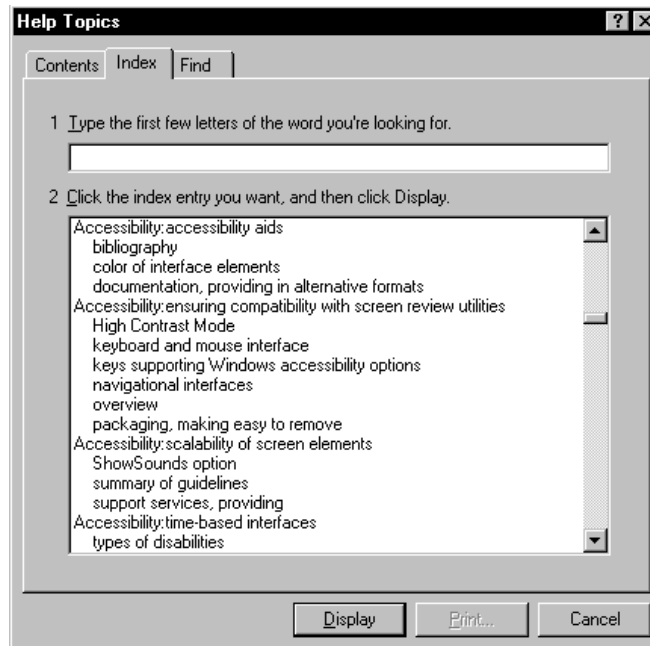


Figure 7: Problems with commas and run-on heads

Sorting problems

You might have some additional problems caused by the effects of alphabetizing and sorting. HTML-based help is supposed to have sorting controls built in, but until then, we have to live with totally automatic sorting.

Articles and prepositions are not ignored in the keyword sort. If you start a keyword with “The,” it will wind up in the “T” section, and your users will not find it. Edit to keep the important words first.

As previously noted, periods in file extensions and quotation marks will also cause sorting problems. Try to leading punctuation in your entries so that they sort properly in the alphabetical scheme.

The worst sorting problem is what I call the dreaded “Mail Merge” problem. This name came from the first set of keywords I created that had this problem, and it comes up in every set I do. You need to check two- or three-word main entries carefully for sorting problems, or you could wind up with results like the ones shown in Figure 8.

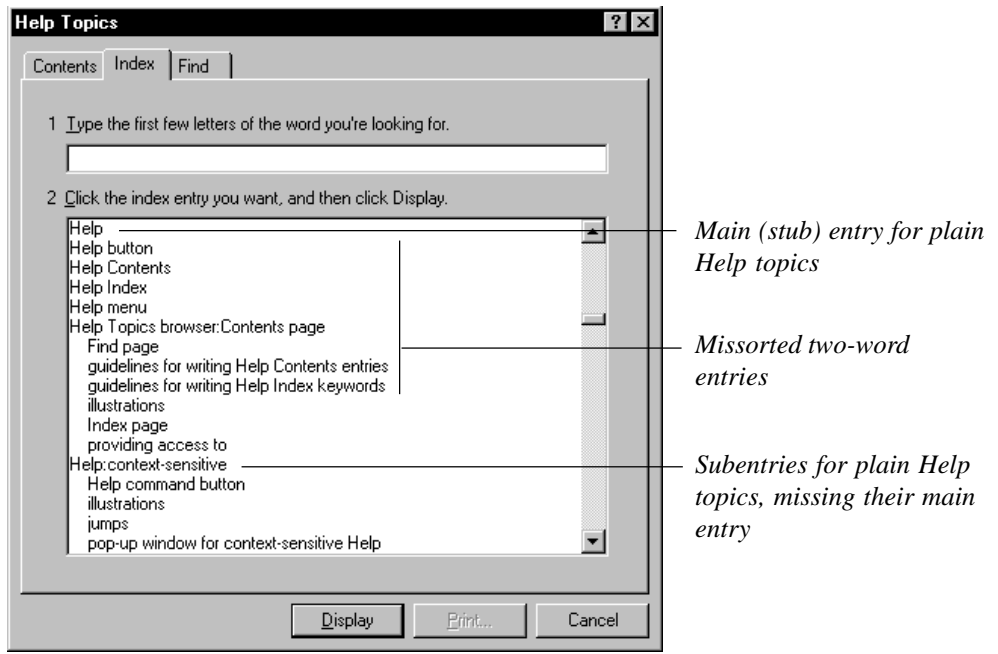


Figure 8: Main entries split from subentries by missorting

Notice how the “Help” main entry is separated from its subentries by the “Help button” and other two-word Help entries. This occurs because WinHelp sorts a space character before the divider character (the colon dividing your main and subentries). What WinHelp is actually doing here is sorting these entries (with all their interior punctuation) in this order:

Help;	—————	Here’s the main entry for Help.
Help button;		————— Here’s some two-word entries, sorted here because WinHelp strips off the end semi-colons, but still considers the colon divider character in sorting. Space characters sort first.
Help contents;		
Help Index;		
Help menu;		
Help Topics browser:Contents page;		
Help Topics browser:Find page;		
Help Topics browser:guidelines for writing Help Contents entries;		
Help Topics browser:guidelines for writing Help Index keywords;		
Help Topics browser:illustrations;		
Help Topics browser:Index page;		
Help Topics browser:providing access to;		
Help:context-sensitive;		————— Here are the subentries for “Help”, and notice how they are now run-on. They’ve lost their main entry.
Help:Help command button;		
Help:illustrations;		
Help:jumps;		
Help:pop-up window for context-sensitive help;		

So how do you fix this problem? What we really want to see is the simple Help topics coming first, followed by the two- and three-word entries. To get this result, add a space before the colon to make the actual entries one space longer. On the main entry, add a space and a superfluous colon. Fortunately, Windows 95 doesn't display the colon in the compile. Here's the revised list, and notice in Figure 9 how it looks compiled.

<p>Help ;: — <i>Here's the main entry for Help, with the extra space and superfluous colon</i></p> <p>Help button;</p> <p>Help contents;</p> <p>Help Index;</p> <p>Help menu;</p> <p>Help Topics browser; — <i>Added to fix the run-on head in this section. Did you notice?</i></p> <p>Help Topics browser:Contents page;</p> <p>Help Topics browser:Find page;</p> <p>Help Topics browser:guidelines for writing Help Contents entries;</p> <p>Help Topics browser:guidelines for writing Help Index keywords;</p> <p>Help Topics browser:illustrations;</p> <p>Help Topics browser:Index page;</p> <p>Help Topics browser:providing access to;</p> <p>Help :context-sensitive;</p> <p>Help :Help command button;</p> <p>Help :illustrations;</p> <p>Help :jumps;</p> <p>Help :pop-up window for context-sensitive help;</p>	<p>————— <i>Here are the subentries for Help, with the extra space added.</i></p>
--	---

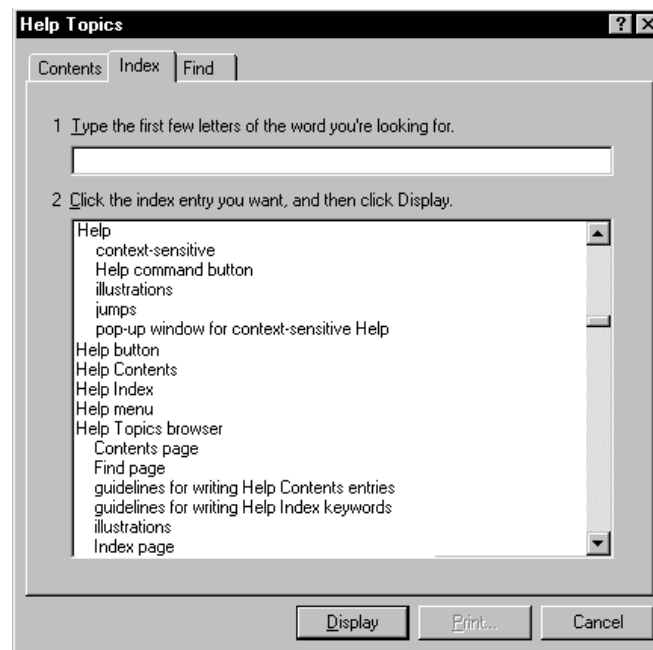


Figure 9: Missorted two-word entries repaired

Notice in Figure 9 how they are now sorting where they belong. There was one extra problem in this section, a run-on entry for the “Help Topics browser” that was lacking a main entry. We added one main stub entry to fix it. Compare this section with Figure 8 and you can see how much easier the list is to read.

You can use this extra space and colon solution to fix Win3.1 keywords, but the colons will show. Windows 95 strips them out so that you can fiddle without it showing.

Some Help authors try tacking a comma on the end of every problem entry instead of the space and colon approach. I don’t recommend doing this, as the sort comes out poorly. The comma will sort after the space character, forcing the simple help entries far below the compound ones, not first where they should be. Our help example would have come out in this order, which is incorrect:

Example:

- Help button
- Help Contents
- Help Index
- Help menu
- Help Topics browser
- Help
 - context-sensitive
 - Help command
 - etc.

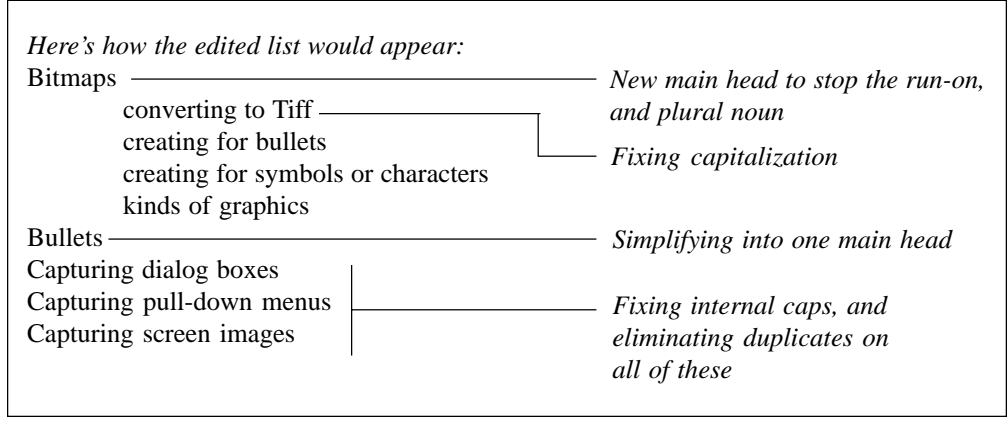
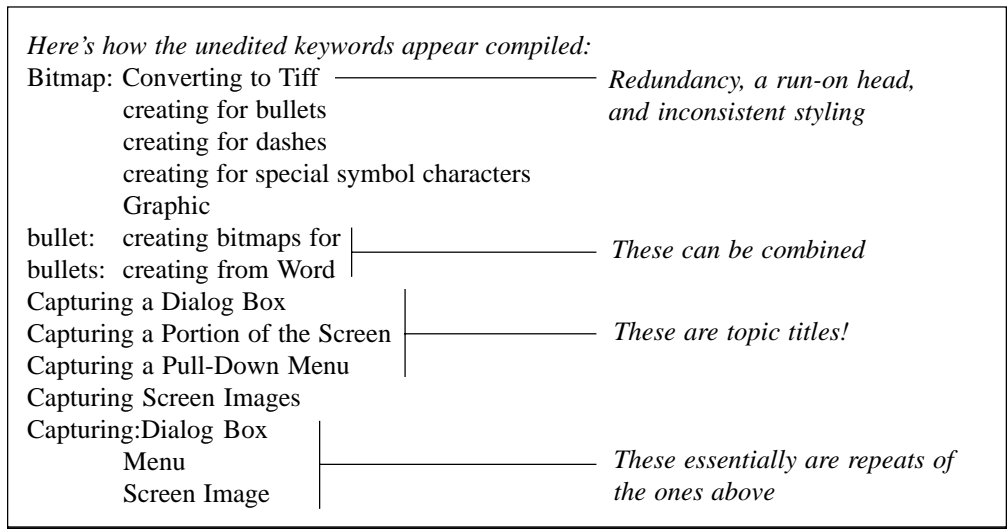
A good edit

Probably the best way to show what happens in a good edit is to show a before-and-after example. Here’s a section of keywords before being editing, and a list of the topics each one brings up:

Keywords	Topics
Bitmap: Converting to Tiff	Converting Bitmaps to Tiffs
Bitmap: creating for dashes	Creating Bitmaps from Symbols
Bitmap: creating for special symbol characters	Creating Bitmaps from Symbols
Bitmap: Graphic	Kinds of Graphics
Bitmap: creating for bullets	Creating Bitmaps from Symbols
bullet: creating bitmaps for	Creating Bitmaps from Symbols
bullets: creating from Word	Converting Word bullets
Capturing a Dialog Box	Capturing a Dialog Box
Capturing a Portion of the Screen	Capturing a Portion of the Screen
Capturing a Pull-Down Menu	Capturing a Pull-Down Menu
Capturing Screen Images	Capturing Screen Images
Capturing:Dialog Box	Capturing a Portion of the Screen
Capturing:Menu	Capturing a Pull-Down Menu
Capturing:Screen Image	Capturing Screen Images

There’s a lot of redundancy and inconsistency in these entries. Let’s see how tight and consistent a good edit can make this section.

Keywords	Topics
Bitmaps <i>(new main stub entry to stop the run-on, and plural noun)</i>	Kinds of Graphics <i>(a general topic)</i>
Bitmaps: converting to Tiff <i>(fixing capitalization)</i>	Converting Bitmaps to Tiffs
Bitmap: creating for dashes <i>(covered by topic below)</i>	Creating Bitmaps from Symbols
Bitmaps: creating for symbols or characters	Creating Bitmaps from Symbols
Bitmaps: kinds of graphics	Kinds of Graphics
Bitmaps: creating for bullets	Creating Bitmaps from Symbols
Bullets <i>(simplifying into one main head and fixing inconsistent capitalization)</i>	Creating Bitmaps from Symbols
Bullets	Converting Word bullets
Capturing dialog boxes <i>(fixing internal caps, and eliminating duplicates on all of these)</i>	Capturing a Dialog Box
Capturing screen images	Capturing a Portion of the Screen
Capturing pull-down menus	Capturing a Pull-Down Menu
Capturing screen images	Capturing Screen Images
Capturing:Dialog Box	Capturing a Portion of the Screen
Capturing:Menu	Capturing a Pull-Down Menu
Capturing:Screen Image	Capturing Screen Images



Ways to achieve consistency

So how on earth do you edit your keywords this consistently? Once they are embedded in the topics, the only view you have of them as an alphabetically-sorted complete list is after compiling, and the only way to edit them is to open up each topic and footnote. Keyword indexers could really use some special tools that would allow them to write, sort, and edit freely in a spreadsheet format, and then quickly and easily embed the keywords in the K footnotes when the keyword structure is set. Unfortunately, most Help development tools ignore this necessary editing stage of keywording.

One beastly way to overcome this lack of keyword indexing tools is to copy and paste keywords into each topic from your spreadsheet, a very time-consuming exercise. Or you could write customized macros to do the embedding once you understand the keywording process.

Whichever method you choose, you still need to edit the raw list of keywords. Work through your keyword list making notes on problem entries, and make repairs. Microsoft's Help Workshop includes report features that can help you with this task. You can use it to generate a report on the compiled keywords in a file, listing the full main and subentries, and the topic name. Figure 10 shows a typical report, and Figure 11 shows the report-generating dialog box.

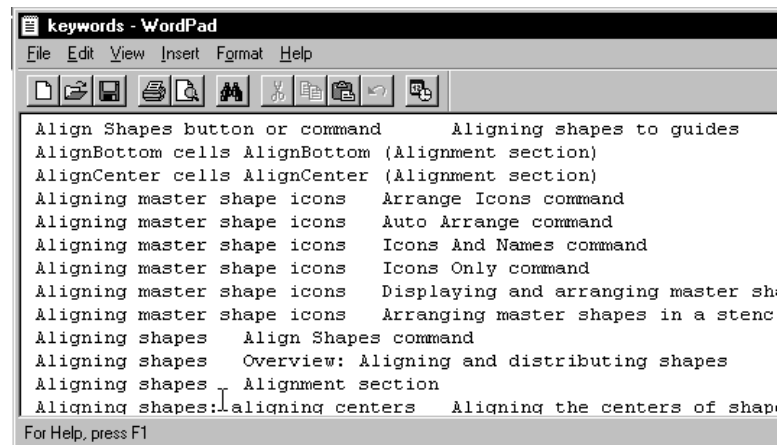


Figure 10: Help Workshop keyword report. Notice you have no way to access the original topic ID numbers, as they disappear in the compile.

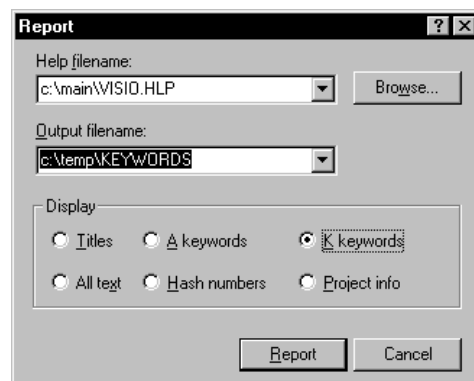


Figure 11: Windows Help Compiler report generation

Working in a spreadsheet

If you decide to take the spreadsheet approach, you actually write your keywords outside the Help files, and embed them as a final step. This has the advantages of allowing you to edit and fine-tune the index until the end, and keeps the Help files free for other needed changes. When the index and the files are ready, you embed the entries all at once, either with a homegrown macro, or agonizingly with cut and paste.

One good method for setting up the spreadsheet is to use one line for each entry, listing the file name, the topic ID, and if you are very energetic, the topic title.

Filename	Topic ID	Keywords:keywords;
Filename	Topic ID	More keywords;
Filename	Topic ID	Yet more keywords;
Filename	Topic ID	Even more keywords;
Filename	Another Topic ID	Keywords;
Filename	Another Topic ID	More Keywords;
Filename	Another Topic ID	Some more Keywords;

Figure 12: Keywords spreadsheet

Note the subentry is listed out fully with the entry, one on each line. You can then sort and write entries in any order you want, comparing and editing as you go without needing to open and close files.

To actually get the entries into the files, you will need to sort by file name and topic ID, and then paste in the entries. To make this easier, save your spreadsheet as text-only with tabs, and open it in Word. Then convert the text to a table. Some fancy searching and replacing will get all your entries for one topic into a usable string for pasting. This gets around the 256 character limitation that spreadsheet cells have.

Filename	Topic ID	Keywords:keywords;
		More keywords;
		Yet more keywords;
		Even more keywords;
Filename	Another Topic ID	Keywords;
		More Keywords;
		Some more Keywords;

You want to get each topic's entries together, ready to paste in.

Figure 13: Working with the entries in a Word table

Filename	Topic ID	Keywords:keywords;More keywords;Yet more keywords;Even more keywords;
Filename	Another Topic ID	Keywords;More Keywords;Some more Keywords

Figure 14: How the keywords should appear for pasting or embedding with macros

Help authoring toolmakers one day will realize that working outside the files is the fastest and most efficient way of indexing in a team setting, as it frees the files for other uses at a critical time. It would be great if one of them would build a set of tools to let indexers work this way easily.

Combining indexes

In Windows 95 you can now combine the indexes for multiple Help files into one display, allowing your users access to many files at once. This is a great advance, but one that also brings design considerations for the indexing. Having a consistent style becomes even more important!

If you decide to combine indexes for a product suite, or meld future update Help files with your current one, you need to know and plan before you start keywording. Otherwise, your keywords might not make sense in the final display.

Note the specificity of these entries for a personal time and contacts management software package.. They should make sense whether they are standing alone, or combined into the final index.

Calendar application keywords	PIM application keywords
Daily calendar updates;	Daily PIM updates;
Monthly calendar updates;	Monthly PIM updates;
Opening: calendar files;	Opening: PIM files;
Printing: calendar views;	Printing: PIM contact lists;
Saving: calendar changes;	Saving: PIM contact list changes;

Planning where the stub entry will go for entries like these gets very difficult. Unless the Help file has a stand-alone topic on “Saving” or “Printing” there’s usually not one good topic for the stub main entry. Often, you will wind up placing stubs in all the topics involved, simply because there is not an appropriate one available.

The Help Compiler Workshop reporting feature can be helpful if you are trying to combine a new set of keywords with an older Help file.

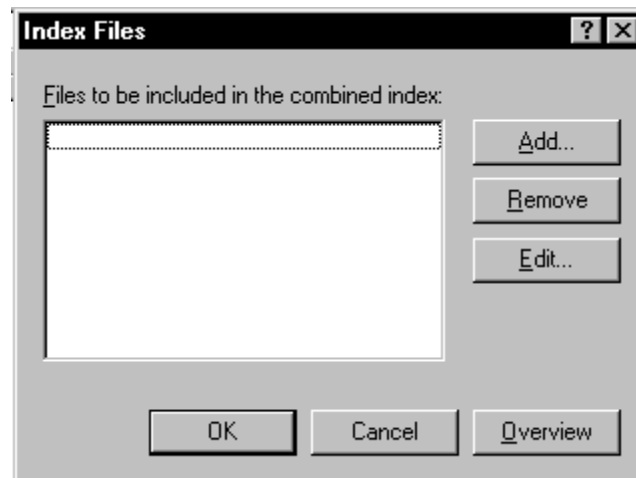


Figure 15: Help Compiler dialog for specifying combined index displays

Scheduling keywords

Scheduling enough time to do keywords well is very hard, due to the amount of information in each topic. One rule of thumb applies: no matter how much time you schedule, the indexing will expand to fill it!

A formula you can use for estimating times and stages for keywording is to multiply the number of screens by 0.04, which will give you the number of 8-hour days. Here's a sample, and if you are somewhat put off by the resulting amount of time, realize that a good set of keywords does indeed require it. There are a lot of factors to be considered.

300 topics at 0.04	12 days
Creating entries (36%)	4.32 days
Editing and embedding entries (24%)	2.88 days
Compiling for review (5%)	0.60 days
Indexer's proofing (5%)	0.60 days
Review (8%)	0.96 days
Incorporate review edits (4%)	0.48 days
Handling revised/updated topics (5%)	0.60 days
Compiling (5%)	0.60 days
Final review and fixes (8%)	0.96 days

If you are pressed for time, some stages of the process can be skipped, as long as you understand the trade-offs. Try not to cut down any times for creating and editing the index, or the compile times. If you hand over the files for indexing in a final state, you can skip the "Handling revised topics and updates" time. If you decide to only have one review, you can skip the "Final review and fixes" stage.

The next wave of Help

Since Microsoft announced that it would be abandoning the WinHelp engine and converting to an HTML-based Help system, pandemonium has reigned. Help tools developers are racing to figure out what the new system entails, and indexers are waiting to see what kinds of indexing issues will arise with the new tools, and what problems will be fixed.

Netscape has released its own version of HTML help tools, and unfortunately, the system does not appear to have any specialized tools to help keywording large projects. The index is evidently straight HTML links, not generated by keywords. This system would be even more labor-intensive than manually editing keywords in WinHelp, as at least WinHelp will alphabetize and combine like entries, allowing multiple destination topics. Third party solutions based on Java tools could help, but we will have to wait and see what is developed.

Microsoft's HTML help tools are not released yet, but some preliminary information is available about the indexing tools.

According to the Microsoft Web site, HTML help will be based on a sitemap standard, and the index will be enabled by an ActiveX object. The cross-references are going to be active, in that they will jump a user to the correct section of the index, not just to the topic they are embedded in. Sorting controls to help resolve a lot of the past sorting problems should be available. The index can contain more than two levels if needed, and the jumps can lead to multiple destinations as well as remote Internet locations. And finally, there will be an HHK index editor tool to help with the task of building the index. Final details about how all these features will work are not resolved yet.

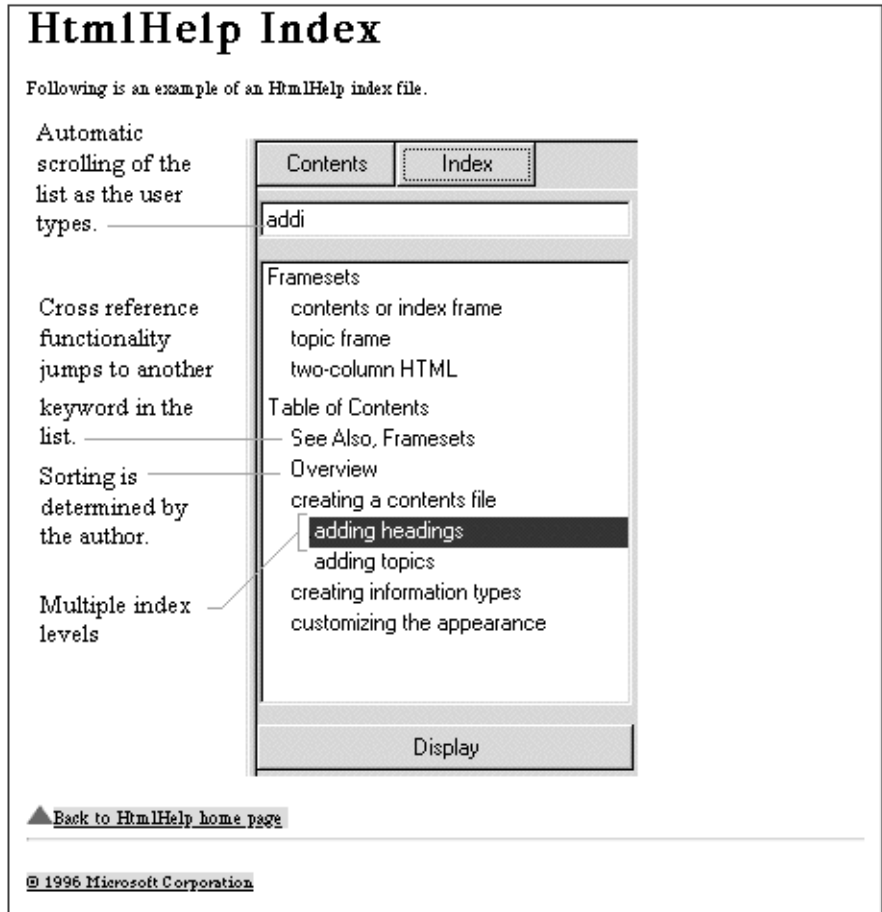


Figure 16: From the Microsoft HTML Help Web site

The benefits of good editing

Is all this work worth it? Users think it is, since they are now trying to find most of their product information online. As I said before, your users find poorly designed keywords just as frustrating to use as you do. Although Windows 95's book-like keyword indexes are a marvelous advance, they can create additional problems for Help writers, since a poorly written set of keywords is almost immediately visible. The indenting structure reveals inconsistencies instantly. It seems paradoxical that a good set of keywords is rarely noticed, but a bad set is cursed daily. The ideal set of keywords allows a user to look through the list, find the topic, and get on with their task, free of distraction or frustration.

References and Further Readings

- American Society of Indexers, *Index Evaluation Checklist*. Mostly geared towards print indexes, but still a valuable guideline. Available from American Society of Indexers, P.O. Box 48267, Seattle, WA 98148-0267, (206) 241-9196
- Anderson, James D., "Indexing Standards: Are They Possible? What Good Are They? Why Bother?" *Proceedings of the American Society of Indexers 25th Annual Meeting*, 1993. Available from American Society of Indexers, P.O. Box 48267, Seattle, WA 98148-0267, (206) 241-9196
- Boggan, Farkas, and Welinske, *Developing Online Help for Windows 95*, ITCP, 1996. (800) 842-3636. Has a good discussion of WinHelp keywording techniques.
- Deaton, Mary and Zubak, Cheryl, *Designing Windows 95 Help*, Que Press, 1996. Another good discussion on working with the WinHelp indexing tools.
- Fillmore, Laura, "Beyond the Back of the Book: Indexing in a Shrinking World," *Key Words: The Newsletter of the American Society of Indexers*, Jul/Aug 1995. A nice discussion of keywording in Cyberspace. Available from American Society of Indexers, P.O. Box 48267, Seattle, WA 98148-0267, (206) 241-9196
- Grech, Christine, "Computer Documentation Doesn't Pass Muster," *PC Computing*, April 1992. Although dated, this survey of what computer users have to say about software documentation and indexes is enlightening.
- Laselle-Ramsay, Inc., "Indexing - No Job for Amateurs," *STC Intercom*, September 1993. Discusses usability testing for technical indexes.
- Lathrop, Lori, "Considerations in Indexing Online Documents," *STC Intercom*, January 1996. Many thanks to Lori for a pre-publication copy of this article. Her lists of qualities of a usable index and items to be keyworded are reused here with mutations and additions.
- Lathrop, Lori, "Indexing with Doc-To-Help: an Overview," *Key Words: The Newsletter of the American Society of Indexers*, Jan/Feb 1995. A overview of converting print documents with indexing into Help files with keywording. Available from American Society of Indexers, P.O. Box 48267, Seattle, WA 98148-0267, (206) 241-9196
- Microsoft HTML Help Web site: <http://www.microsoft.com/workshop/author/htmlhelp>
- Mulvany, Nancy M., *Indexing Books*, University of Chicago Press, 1994. Understanding good print indexing leads to good keywording, and this is one of the best books available on the subject.
- Netscape HTML Help Web site: <http://home.netscape.com/eng/help>
- Winhlp-L mailing list and Web site: <http://www.interlog.com/~gbyrnes/winhelp.html>