For Microsoft Word for Windows™

Doc-To-Help®

Version 1.5

By WexTech Systems, Inc.

Copyright © 1991-1993 WexTech Systems, Inc. All rights reserved.

Microsoft is a registered trademark and Microsoft Excel, Word for Windows, MS Draw and Windows are trademarks of Microsoft Corporation. Word Perfect is a registered trademark of Word Perfect Corporation. MultiMate is a registered trademark of Ashton Tate Corporation. Wordstar is a registered trademark of Micropro Inc. Doc-To-Help is a registered trademark and Quicture is a trademark of WexTech Systems, Inc. Scrapbook+ is a trademark of Eikon Systems, Inc. Hotshot Graphics is a trademark of SymSoft, Inc. CompuServe is a registered trademark of CompuServe, Inc.

This publication was created using Doc-To-Help and Microsoft Word for Windows version 2.0c. Screen shots were captured using Eikon Scrapbook+ and SymSoft Hotshot Graphics. Screen shots were offloaded using WexTech Quicture and annotated using MS Draw. Camera-ready copy was produced using a LaserMaster WinJet controller.

Design and Implementation: Julianne Sharer and Ted Epstein

Documentation: Steve Wexler, Judy Faber and Paul Neshamkin

Editing and Index: Deborah Finn

Drawings: Andre Liu

Additional Programming: Brett Foster and Guy Gallo

WexTech Systems, Inc.
310 Madison Avenue, Suite 905
New York, NY 10017
(212) 949-9595

Technical Support: (212) 949-0093

Contents

Overview 1

Installation 5

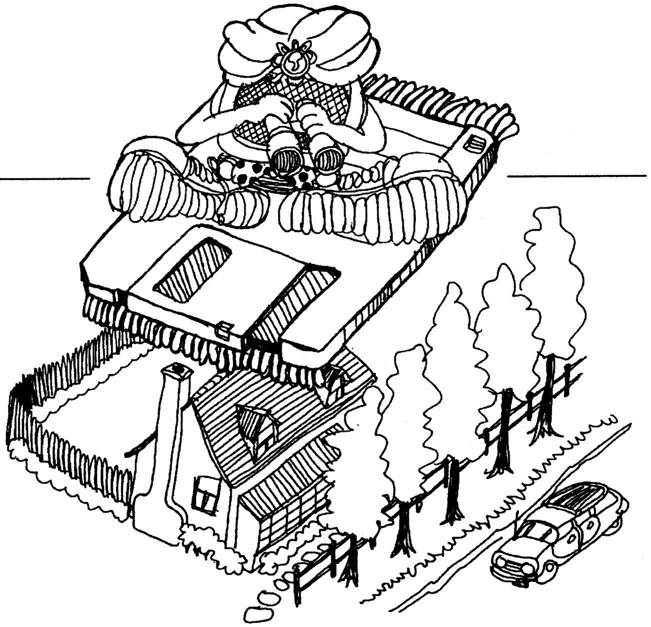
Tutorial/Basics 17

Displaying Chapter Numbers with Page Numbers 75

Converting Existing Documents 83

Advanced Topics 95

Overview



Overview

Introduction

It would be great if every new software package were intuitive and self-explanatory so that anyone could learn how to use it without training or picking up the manual. Unfortunately (or fortunately, for the professional who makes his or her living writing manuals and online help), very few software packages fall into this category. Indeed, the acceptance of your product or application is contingent upon the user's ability to use the product effectively; and learning to use the product effectively without special training or relying on telephone support depends on a good user manual and good online help.

Why People Don't Read the Manual

Studies have shown that a startling number of people never pick up a user manual. This really isn't all that surprising, as most manuals are written badly and are a blight to the eyes. As for online help, most people will use it — if they've been trained to use it and if it is available. However, only the brave have been likely to tackle developing online help as creating help used to be a daunting and time-consuming task.

We have designed Doc-To-Help so that anyone — whether an applications programmer, technical writer or human resources professional — can write clear, concise and attractive documentation and deliver professional online help. Without tears.

Uses for Doc-To-Help

While we originally designed Doc-To-Help with the developer/ documentation specialist in mind, Doc-To-Help has been put to use in areas that we never anticipated. These uses include employee handbooks, machine assembly and mainframe access procedures. Basically, anything you might want to document and convert to Windows Help can be handled by Doc-To-Help.

Who Should Use Doc-To-Help?

The Novice

RTF stands for "Rich Text Format." You create online help by preparing a document in this file format, which is then compiled into an HLP file.

Doc-To-Help has been designed so that a person with modest Word for Windows skills can produce professional-quality documentation and deliver rich online help without spending days meticulously annotating the requisite RTF file. On the "manual" side, Doc-To-Help takes care of styles, page layout, floating footers, callouts, cross references and indexing. On the "Help" side, Doc-To-Help takes care of generating the precisely formatted RTF file. You need never "look under the hood" to deal with the complexities of setting context strings, keywords, browse sequences, and so on.

The Expert

While the styles are available for you to examine and change, the macros associated with the custom menus are not. You can add new macros, but you cannot edit existing ones.

Basically, Doc-To-Help will allow you to produce better work faster. Doc-To-Help is also extremely flexible. While we have provided three sets of templates — one of which will probably meet your needs — our open architecture will allow you to change styles and fonts, if you feel they need to be changed. And while indexing will never be a pleasure, Doc-To-Help makes the process of generating a complete index considerably less painful.

If you've ever worked with the Microsoft Help compiler before, you know that producing the necessary RTF file and accompanying project file is, at best, tedious. Doc-To-Help will generate a complete, fully-formatted RTF file and project file for you; however, if you want to edit the file—perhaps to add cross reference jumps that Doc-To-Help could never have anticipated—you can certainly do so.

How This Manual Is Organized

This manual is divided into thirteen major sections.

Installation

How to install Doc-To-Help on your computer and descriptions of what each file on the program disks is for.

Tutorial/Basics

A tutorial that takes you through the steps needed to create a manual and accompanying Windows online help. The tutorial takes approximately 30 minutes to complete.

How Doc-To-Help Turns Manuals into Online Help

A detailed discussion of how Doc-To-Help converts manuals into online Help files.

Working with Long Documents

More on topics discussed in the Tutorial/Basics section and details on how to build manuals from separate files.

Indexing

An in-depth discussion of all the indexing features.

Displaying Chapter Numbers with Page Numbers

A discussion of formatting footers, tables of contents and indexes to include both the chapter and page number.

Converting Existing Documents

Methods for making documents created outside of Doc-To-Help or in other word processors into Doc-To-Help manuals or Help files. A discussion of how to assimilate existing help files into a Doc-To-Help document.

Advanced Topics

How to mark text for the manual/help only, how to insert cross references and page references, how to insert hypertext links, how to format source code, how to print a list of context strings and context string mappings. Information about multiline captions, graphics, modifying the appearance of the manual and Help screens, formatting source code, symbols and special characters, copying manuals and user-defined WordBasic macros.

Using the Segmented Hypergraphics Editor

How to use the Microsoft Segmented Hypergraphics Editor (SHED) to create multiple hot spots on a single graphic.

Help Macros

How to make more sophisticated help files by adding Help Macros.

Troubleshooting

Suggestions for solving problems with large files and memory management.

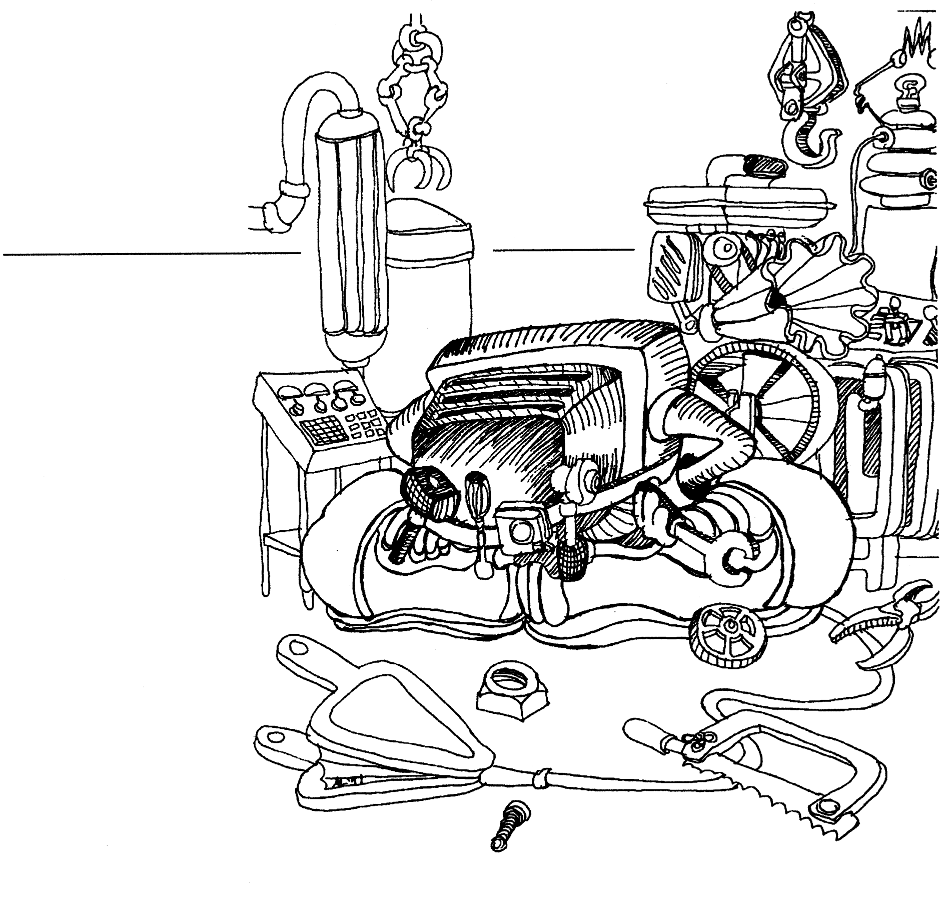
Reference

Template and style examples, a list of all menu commands, style definitions and shortcut keys.

Appendix: Accessing Help

How to access Help files from Program Manager, Word for Windows, Excel, Visual Basic and C. Includes detailed information on how to create context-sensitive help in Word for Windows, Excel, Access, Visual Basic and C.

Installation



Installation

What You Will Need

To use Doc-To-Help, you will need the following.

- Microsoft Windows release 3.0 or later
- At Least 2MB of memory (4MB is preferable)
- Microsoft Word for Windows, version 2.0 or later

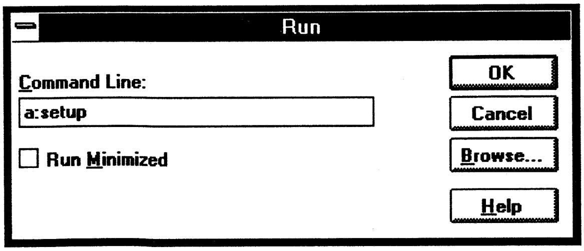
The Doc-To-Help diskettes contain a setup utility that will install the Doc-To-Help templates, the tutorial files, the Microsoft Help Compilers and several other utility files. For this example, let's assume that you are installing Doc-To-Help from drive **A.**

Installation: US Version vs. International Version

Doc-To-Help employs a different setup program depending on whether you have purchased the US or International version of Doc-To-Help. Please follow the instructions that are applicable to the version you are using. Also, if you will be installing Doc-To-Help on a network, please read "If You're Using Windows on a Network" on page 14.

To Install Doc-To-Help (US Version)

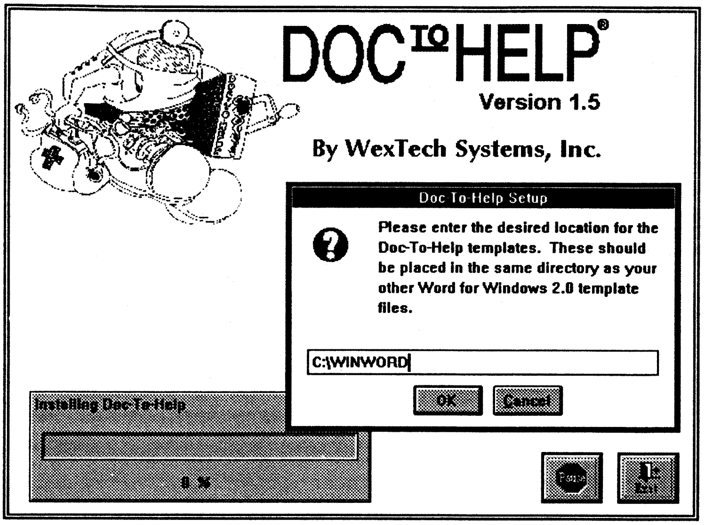
1. Place Disk 1 of the two Doc-To-Help diskettes in the floppy disk drive.
2. From the Program Manager, click **File** and choose **Run**. (You can also do this from the File Manager.)
3. When the dialog box appears, type **a:setup**, as shown below.



Command Line: a:setup

□ Run Minimized

4. Click **OK**. After several seconds your screen should look like the one shown below.



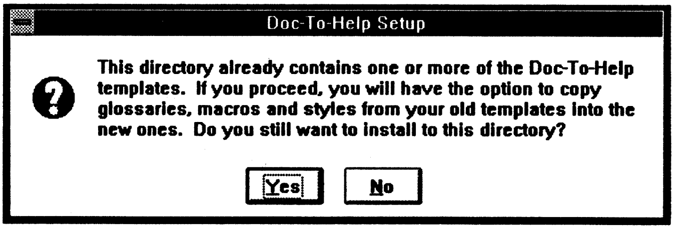
Doc-To-Help Setup

Please enter the desired location for the Doc-To-Help templates. These should be placed in the same directory as your other Word for Windows 2.0 template files.

C:\WINWORD

5. If what is in the box is not correct, type the drive and directory name where your Word for Windows 2.0 templates are stored, and click **OK**.

5a. If the directory specified in step 5 contains earlier versions of the Doc-To-Help templates, the following dialog box will appear:



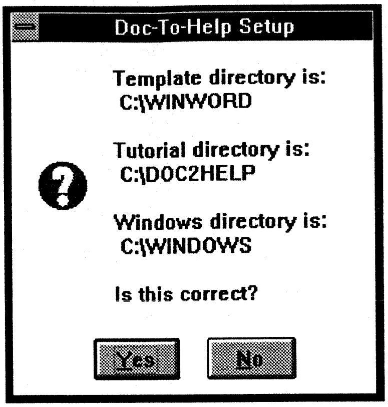
Doc-To-Help Setup

This directory already contains one or more of the Doc-To-Help templates. If you proceed, you will have the option to copy glossaries, macros and styles from your old templates into the new ones. Do you still want to install to this directory?

Click **Yes** to continue.

6. When prompted, type the drive and directory where you want the setup utility to copy files used for the tutorial and click **OK**.

7. When prompted, type the drive and directory where your Windows files are stored and click **OK**. A dialog box similar to the following will appear.



Template directory is:
C:\WINWORD

Tutorial directory is:
C:\DOC2HELP

Windows directory is:
C:\WINDOWS

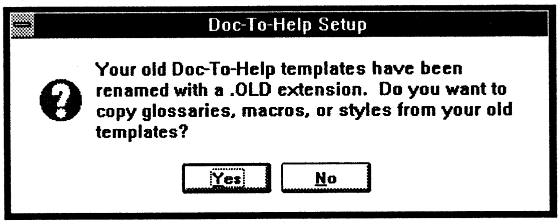
Is this correct?

8. If the path names are correct, click **Yes**. When prompted, insert Disk 2 of the two Doc-To-Help diskettes.

The setup routine does not automatically transfer custom headers, footers or page setups from the old templates into the new templates.

If you want to copy a footer, open up the old template that contains the footer, view and, copy the footer, open the new template, view and select the new footer, then paste over the footer.

8a. If the directory specified in step 5 contains earlier versions of the Doc-To-Help templates, the following dialog box will appear:

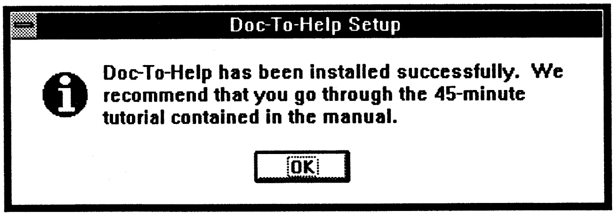


Your old Doc-To-Help templates have been renamed with a .OLD extension. Do you want to copy glossaries, macros, or styles from your old templates?

Click Yes if the old Doc-To-Help templates contain any glossaries, macros or styles you want to copy into the new templates.

9. After the setup utility has finished installing Doc-To-Help, the following dialog box will appear.

Taking the tutorial is highly recommended.

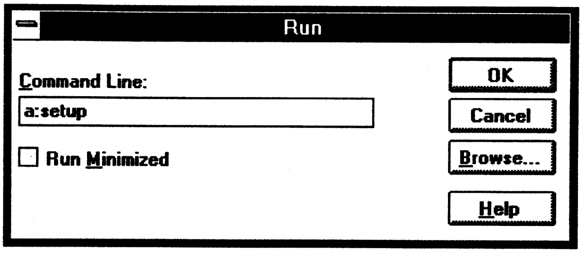


Doc-To-Help has been installed successfully. We recommend that you go through the 45-minute tutorial contained in the manual.

10. Click **OK**.

To Install Doc-To-Help (International Version)

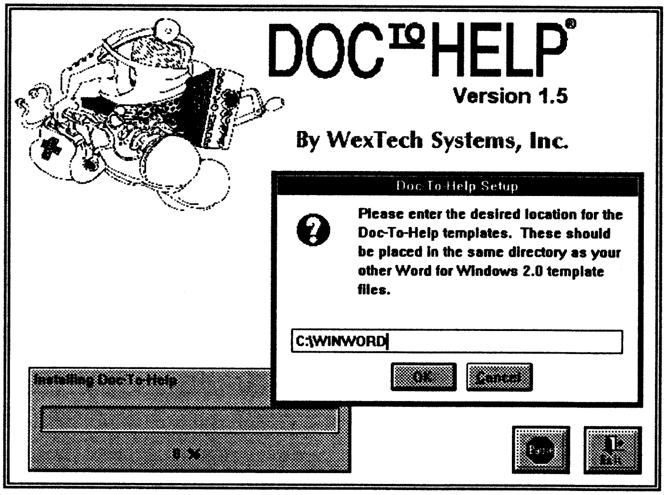
1. Place Disk 1 of the two Doc-To-Help diskettes in the floppy disk drive.
2. From the Program Manager, click **File** and choose **Run**. (You can also do this from the File Manager.)
3. When the dialog box appears, type **a:setup**, as shown below.



Command Line: a:setup

□ Run Minimized

4. Click **OK**. After several seconds your screen should look like the one shown below.



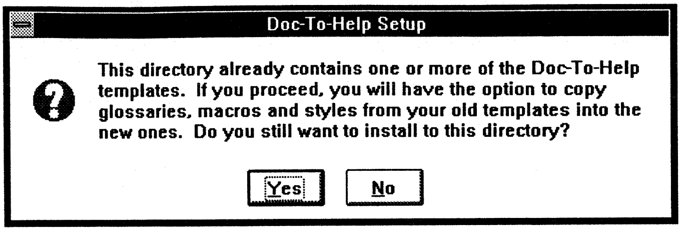
Please enter the desired location for the Doc-To-Help templates. These should be placed in the same directory as your other Word for Windows 2.0 template files.

C:\WINWORD

WordBasic cannot handle diacritical marks in path names. Make sure that all of the paths you specify do not contain accented characters.

5. If what is in the box is not correct, type the drive and directory name where your Word for Windows 2.0 templates are stored, and click **OK**.

5a. If the directory specified in step 5 contains earlier versions of the Doc-To-Help templates, the following dialog box will appear:

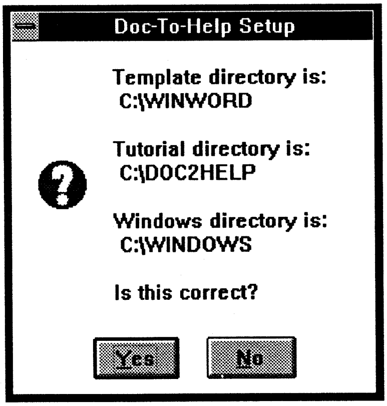


This directory already contains one or more of the Doc-To-Help templates. If you proceed, you will have the option to copy glossaries, macros and styles from your old templates into the new ones. Do you still want to install to this directory?

Click **Yes** to continue.

6. When prompted, type the drive and directory where you want the setup utility to copy files used for the tutorial and click **OK**.

7. When prompted, type the drive and directory where your Windows files are stored and click **OK**. A dialog box similar to the following will appear.



Template directory is:
C:\WINWORD

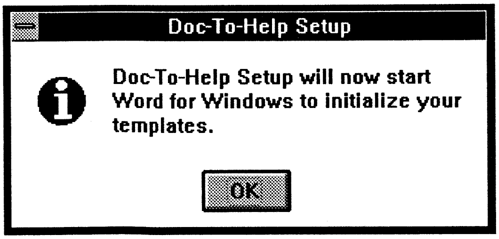
Tutorial directory is:
C:\DOC2HELP

Windows directory is:
C:\WINDOWS

Is this correct?

8. If the path names are correct, click **Yes**. When prompted, insert Disk 2 of the two Doc-To-Help diskettes.

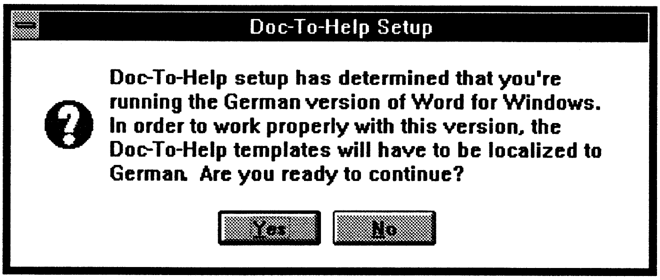
9. After the setup program has copied files to your system, the following dialog box will appear.



Doc-To-Help Setup will now start Word for Windows to initialize your templates.

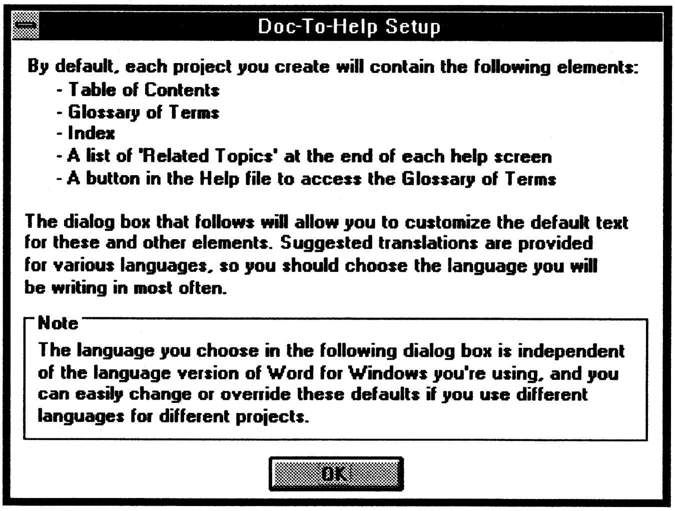
At this point, files have been copied to your system, but the Doc-To-Help templates will not function properly as they need to be modified to work with the version of Word for Windows you are using. In addition to modifying the templates so that they work with your version of Word for Windows, the setup program gives you the opportunity to change the boilerplate text and other text defaults that will appear in your manuals and help files.

10. Click **OK**. The setup program will start Word for Windows, load the file D2H_UTIL.DOC, and display a dialog box similar to the following:



Doc-To-Help setup has determined that you're running the German version of Word for Windows. In order to work properly with this version, the Doc-To-Help templates will have to be localized to German. Are you ready to continue?

11. Click **Yes**. The following explanatory dialog box will appear:



By default, each project you create will contain the following elements:

- Table of Contents
- Glossary of Terms
- Index
- A list of 'Related Topics' at the end of each help screen
- A button in the Help file to access the Glossary of Terms

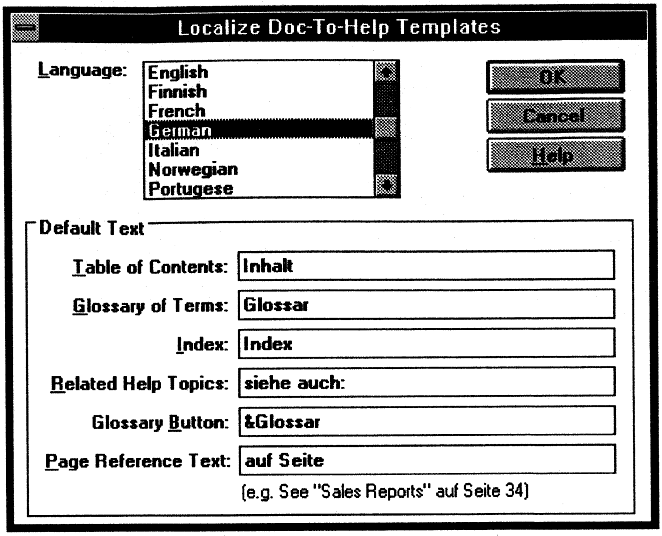
The dialog box that follows will allow you to customize the default text for these and other elements. Suggested translations are provided for various languages, so you should choose the language you will be writing in most often.

Note

The language you choose in the following dialog box is independent of the language version of Word for Windows you're using, and you can easily change or override these defaults if you use different languages for different projects.

12. Click **OK**. The Localize Doc-To-Help Templates dialog box will appear, as shown below:

The localize dialog box allows you to specify default text applicable to the language you will be writing in most often, not the language version of Word for Windows you are using. For example, you may be using the US version of Word for Windows, but usually write in French. If this is the case, you should select French from the Language list box, and then customize the default text entries to suite your needs. The default settings will appear in boilerplate template text, Doc-To-Help-created page references and heading text for subtopics in your help files.



Localize Doc-To-Help Templates

Language: English, Finnish, French, **German**, Italian, Norwegian, Portugese

Default Text

Table of Contents: Inhalt

Glossary of Terms: Glossar

Index: Index

Related Help Topics: siehe auch:

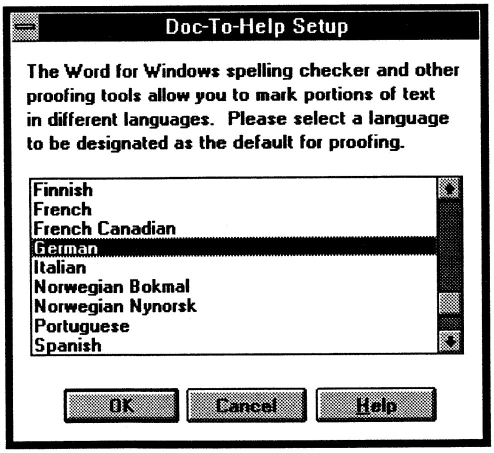
Glossary Button: &Glossar

Page Reference Text: auf Seite

(e.g. See "Sales Reports" auf Seite 34)

Please note that the default text specified here is independent of the version of Word for Windows you are using. This means that if you write in German, but use the US version of Word for Windows, you can use German text defaults without upsetting anything.

13. Fill in the dialog box as desired and click **OK**. The following dialog box will appear:



Doc-To-Help Setup

The Word for Windows spelling checker and other proofing tools allow you to mark portions of text in different languages. Please select a language to be designated as the default for proofing.

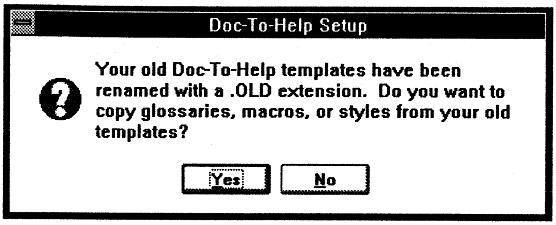
Finnish, French, French Canadian, **German**, Italian, Norwegian Bokmal, Norwegian Nynorsk, Portuguese, Spanish

14. Select the spelling checker you use most often and click OK. The setup program will modify the seven Doc-To-Help templates.

The setup routine does not automatically transfer custom headers, footers or page setups from the old templates into the new templates.

If you want to copy a footer, open up the old template that contains the footer, view and copy the footer, open the new template, view and select the new footer, then paste over the footer.

14a. If the directory specified in step 5 contains earlier versions of the Doc-To-Help templates, the following dialog box will appear:



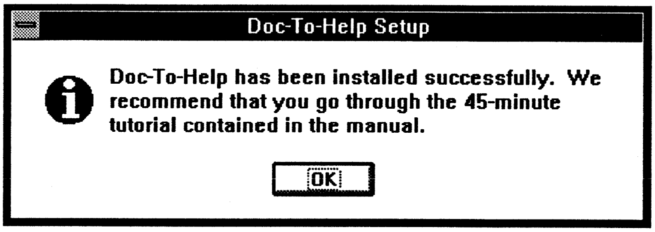
Doc-To-Help Setup

Your old Doc-To-Help templates have been renamed with a .OLD extension. Do you want to copy glossaries, macros, or styles from your old templates?

Click Yes if the old Doc-To-Help templates contain any glossaries, macros or styles you want to copy into the new templates.

Taking the tutorial is highly recommended.

15. After the setup utility has finished installing Doc-To-Help, the following dialog box will appear.



Doc-To-Help Setup

Doc-To-Help has been installed successfully. We recommend that you go through the 45-minute tutorial contained in the manual.

Using Doc-To-Help with More than One Language Version of Word for Windows

It is possible (and in fact, rather easy) to use Doc-To-Help with more than one language version of Word for Windows. Basically, you need to:

- Run the Doc-To-Help setup program for each version of Word for Windows you are using.
- Tell Word for Windows where the Doc-To-Help templates can be found by changing the DOT-PATH setting in your WIN.INI file.
- Tell Doc-To-Help which language version you are using through the Tools Doc-To-Help Options dialog box.

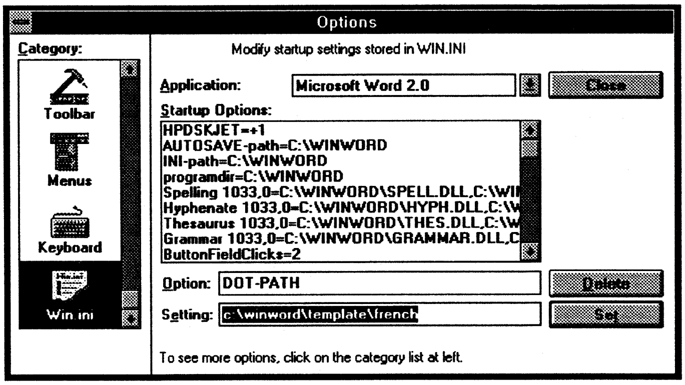
To Install Doc-To-Help for Use with Different Language Versions of Word for Windows

1. Create directories to house different sets of templates for the versions of Word for Windows you will using. For example, if you will be using both the French and German versions of Word for Windows, create two directories, c:\winword\template\french and c:\winword\template\german.

2. Run the Doc-To-Help setup program for each language version of Word you will be using, installing the Doc-To-Help templates in the directories created in Step 1, and localizing each set of templates, as prompted by the setup program.

To Use Doc-To-Help with a Particular Version of Word for Windows

1. Change the DOT-PATH in your WIN.INI file so that Word for Windows knows where to find the Doc-To-Help templates. You can change the DOT-PATH setting through the Tools Options WIN.INI dialog box (an example is shown below).



Modify startup settings stored in WIN.INI

Application: Microsoft Word 2.0

Startup Options:

```
HPDSKJET=+1
AUTOSAVE-path=C:\WINWORD
INI-path=C:\WINWORD
programdir=C:\WINWORD
Spelling 1033,0=C:\WINWORD\SPELL.DLL,C:\WI
Hyphenate 1033,0=C:\WINWORD\HYPH.DLL,C:\W
Thesaurus 1033,0=C:\WINWORD\THES.DLL,C:\W
Grammar 1033,0=C:\WINWORD\GRAMMAR.DLL,C
ButtonFieldClicks=2
```

Option: DOT-PATH

Setting: c:\winword\template\french

To see more options, click on the category list at left.

2. With a Doc-To-Help document loaded, click on **Tools** and **Choose Doc-To-Help Options**. The Doc-To-Help Options dialog box will appear.

3. Click **Word for Windows Version**. The Word for Windows Version dialog box will appear.

4. Select the language version of Word for Windows you will be using and click **OK**.

5. When the Doc-To-Help Options dialog box reappears, click **OK**.

D2H_GLBL.INI contains global settings such as WinWord language version and indexing preferences.

Employing the Tools Doc-To-Help Options dialog box to change your Word for Windows version just changes the Language setting in your D2H_GLBL.INI file (which can be found in the Windows directory). If you want, you can bypass steps two through five listed above and make the modifications to D2H_GLBL.INI directly using the values in the table shown below.

Language	Language Setting
English	Language=0
Italian	Language=100
Swedish	Language=200
Dutch	Language=300
Norwegian	Language=400
Danish	Language=500
Finnish	Language=600
German	Language=700
French	Language=800
Spanish	Language=900
Portuguese	Language=1000

If You're Using Windows on a Network

The setup program will not work properly if your Windows directory is mapped to a false root. For example, let's suppose your personal drive is mapped as a root to **o:\users\yourname,** but appears to you as **n:\**. The setup program will have a problem if you type **n:\** as your Windows directory. If this is the case, when prompted for the name of your Windows directory you can either

- type **o:\users\yourname**, or
- type a shared directory that's mapped to a search drive.

What's on the Disks

The setup program installs the following files:

Files copied to the Word for Windows template directory

File	Description
D2H_NORM.DOT	Master template for producing manuals on standard size paper (letter size in US version, A4 size in international version).
D2H_IN.DOT	Template used to produce inside chapters for standard manuals. This template is used when you want to divide the chapters of your manual into separate files.
D2H_SIDE.DOT	Master template for producing manuals that use sideheads. Sidehead style places headings in the left margin and places body text immediately to the right (see page 196).
D2H_INSD.DOT	Template used to produce inside chapters for sidehead style manuals.
D2H_SMAL.DOT	Master template for producing manuals on 7-inch by 9-inch paper. (Prints on standard size paper with crop marks.)
D2H_INSM.DOT	Template used to produce inside chapters for manuals printed on 7-inch by 9-inch paper.
D2H_HELP.DOT	Template used for producing Help files. Doc-To-Help will activate this template when you convert your manual into online help.

Files copied to the Windows directory

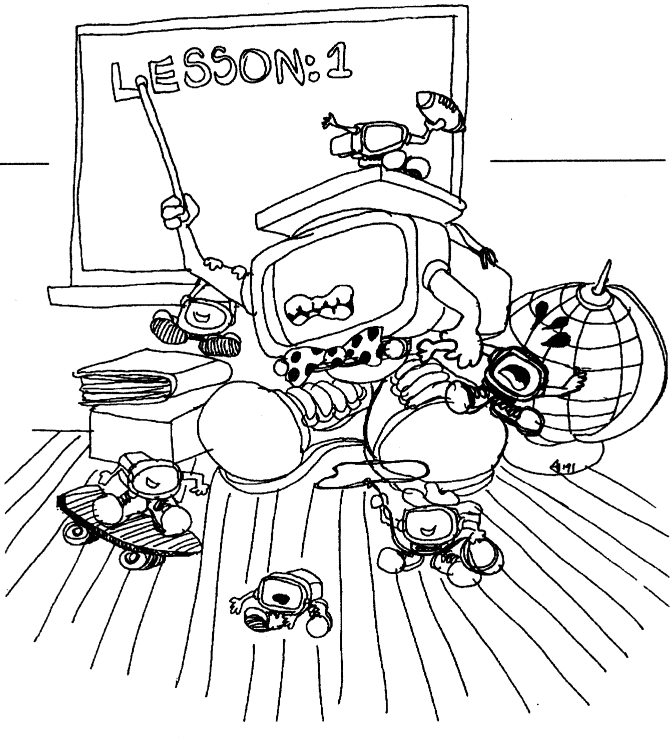
File	Description
D2HELPER.EXE	Program that insures that DOC2HELP.DLL works properly with Word for Windows. D2HELPER.EXE loads automatically whenever you use Doc-To-Help.
DOC2HELP.DLL	Dynamic Link Library used by Doc-To-Help.
HC.EXE	Microsoft Help Compiler for Windows 3.0
HC31.EXE	Microsoft Help Compiler for Windows 3.1
HC31.ERR	Error message file for HC31.EXE
HCP.EXE	Protected mode Help Compiler for Windows 3.1
HCP.ERR	Error message file for HCP.EXE
WINHELP.EXE	Windows 3.1 Help engine.
DOC2HELP.HLP	Doc-To-Help online Help file.
SHED.EXE	Microsoft Segmented Hypergraphics Editor
SHED.HLP	Help file for SHED.EXE

Files copied to the tutorial directory

File	Description
PART1.DOC	Sample file used in tutorial.
PART2.DOC	Sample file used in tutorial.
PETS.WMF	Sample picture used in tutorial.

For samples of what the manuals look like, see "Doc-To-Help Templates" on page 195.

Tutorial/Basics



Tutorial/Basics

Creating a New Manual

Prerequisites and Overview

We are assuming that you are comfortable using Word for Windows. If you are a little uncertain as to how you create files and format text in Word, you'll probably want to take the Word tutorial before proceeding. You access this tutorial by choosing **Learning Word** from the **Help** menu.

In this section we will explore, step-by-step, the procedures needed to create a manual and accompanying online help. This "guided tour" will take about 45 minutes to complete and will expose you to the major features of Doc-To-Help.

Before continuing, do the following:

- Create a new directory to hold the file(s) that will comprise the practice manual.
- Load Microsoft Word for Windows 2.0 or later.

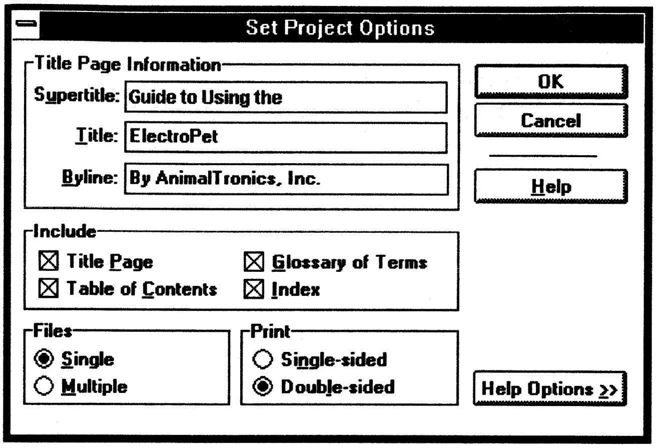
For the next series of example, you will create a simple manual for a fictitious product called the "ElectroPet." It will contain a title page, a table of contents, two main chapters, a glossary of terms and an index.

To Create a New Manual

1. From the **File** menu, choose **New**. The **File New** dialog box will appear. Do **not** change the option button to **Template**.
2. From the list of available templates, choose **D2H_NORM** and click OK. (Samples of the Doc-To-Help templates can be found in "Doc-To-Help Templates" on page 195.) The Document Preferences dialog box will appear.

The Supertitle (Guide to Using the) and the Title (ElectroPet) will be inserted in the Title text box in the File Summary Info dialog box. The document title, as well as each chapter name, will appear in the footer of your document.

3. Fill in the dialog box as shown below.



Set Project Options

Title Page Information

Supertitle: Guide to Using the

Title: ElectroPet

Byline: By AnimalTronics, Inc.

Include

☒ Title Page ☒ Glossary of Terms

☒ Table of Contents ☒ Index

Files

● Single

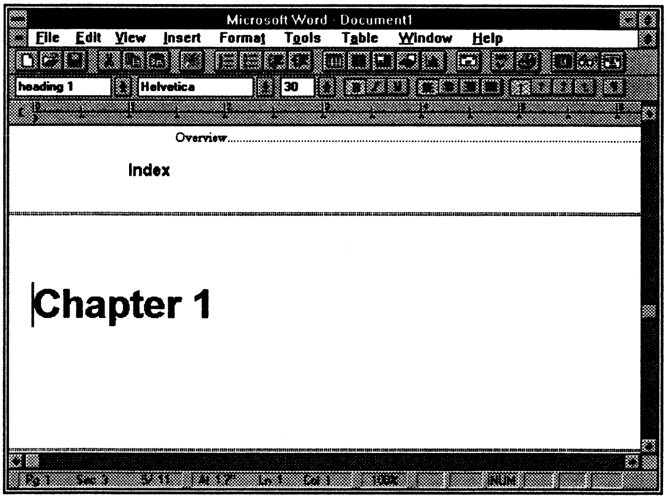
○ Multiple

Print

○ Single-sided

● Double-sided

4. Click **OK**. Doc-To-Help will take you to the first chapter of your new document, as shown below.



Microsoft Word - Document1

Overview..

Index

Chapter 1

Saving Your Work

Each manual you write should be saved in a different directory as there can be only one DOC2HELP.INI file per directory.

Before you go on, you should save your work. Saving your work automatically enters the Title in the File Summary Info dialog box and creates a file called DOC2HELP.INI in the directory where you save this manual. DOC2HELP.INI is a text file that contains information Doc-To-Help needs to correctly format your manual and Help file. If you copy the manual to a different directory, make sure you copy the DOC2HELP.INI file as well.

To Save the File

1. From the **File** menu, choose **Save.**
2. Type the name you want to give this file, in this case **SAMPLE.DOC**.
3. Choose the drive and directory where you want to save the manual and click **OK**.

Examining the File

To examine the file, move to the beginning of the document by pressing CTRL+HOME and choose **Print Preview** from the **File** menu. You can examine the pages of the manual using the scroll arrows or by pressing the PAGE DOWN key.

The document is divided into six sections, separated by section breaks. Do not delete these section breaks manually.

Title Section

*Note: The **Format Set Project Options** command is not available when you are in **Print Preview** mode.*

This section contains the title page and copyright/trademark page. The title page includes a supertitle, title and byline as well as a picture place holder. Text on the title page should be modified using the **Format Set Project Options** command. The picture placeholder can be used to place a graphic on the title page. It can be edited using standard Word commands.

If you don't want a title page in your document, make sure Title Page is not selected in the Format Set Project Options dialog box.

Table of Contents

The second section contains a table of contents field for producing a table of contents based on heading styles 1-3. The table of contents can be updated using the Update Fields command (function key F9) or when you print your document. The footer for this section contains the document title and chapter name. Page numbers are formatted using Roman numerals (i, ii, iii, and so on).

If you don't want a table of contents, make sure Table of Contents is not selected in the Format Set Project Options dialog box.

Chapter 1

The first chapter has a text place holder called "Chapter 1" formatted in Heading 1 style. You can change the place holder text but you must leave the chapter heading formatted in Heading 1 style. (We'll discuss styles in a moment.)

Page numbers have been set to start at 1, formatted using Arabic numerals (1, 2, 3, and so on). If you want to change the format of page numbers, you can use **View Header/Footer** for each section in your document. For information on putting chapter numbers in the footer, see "Displaying Chapter Numbers with Page Numbers" on page 75.

Chapter 2

The second chapter contains a text place holder called "Chapter 2" also formatted in Heading 1 style. If your manual will be printed on both sides of the page, page numbers have been set to begin at the next *odd* page after the previous chapter's last page. If your manual will be printed on only one side, page numbers have been set to begin at the next page after the previous chapter's last page.

Glossary of Terms

The next section will contain the glossary of terms. As you compile your documentation, you will undoubtedly introduce concepts and terms which will be new to your readers. Whenever you introduce a new term or concept, you should add it to the Glossary of Terms using the Insert Glossary Term command.

If you don't want your document to contain a glossary of terms, make sure Glossary of Terms is not selected in the Format Document Preferences dialog box. Pages will be numbered the same as they are in Chapter 2.

Index

The final section of the manual will contain the document's index, which, unless the document is very short, you better have a darn good reason for omitting. A good index is essential to a useful manual. Pages will be numbered the same as they are in Chapter 2.

Before continuing, cancel out of Print Preview.

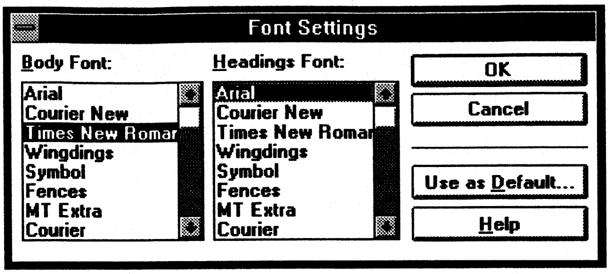
Setting the Fonts

__Arial__ and __Times New Roman__ are TrueType fonts available with Windows 3.1. Windows 3.0 users will need to change the fonts to match their printer.

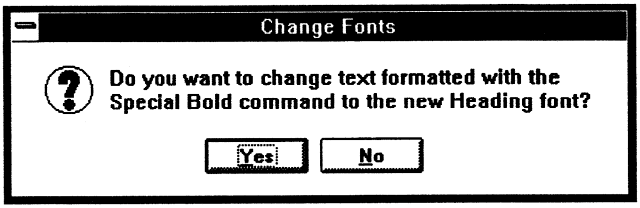
By default, Doc-To-Help templates have been formatted so that headings are in Arial bold and body text is in Times New Roman. While this combination will certainly produce an attractive document, you may decide you want to work with different type faces. The easiest way to change the heading and body fonts is to use the **Format Change Fonts** command.

To Change Fonts

1. From the **Format** menu, choose **Change Fonts**. The following dialog box will appear:



2. Select the fonts you want to use and click **OK**. Clicking Use as Default will alter the underlying template so that all new documents based upon this template will use the fonts you choose. If you change the heading font, the following dialog box will appear.



Do you want to change text formatted with the Special Bold command to the new Heading font?

For now, just click **Yes**. For more information, see “Special Bold” on page 29 and “Format Change Fonts” on page 224.

Replacing the Chapter 1 Place Holder

If you are using a non-English version of Word for Windows, typing "S3" may not work as some versions of Word use a different letter to designate a section.

Note: Replacing the place holder in this fashion will not work if "Typing Replaces Selection" is off in the ***Tools Options General*** *dialog box.*

You're now ready to start writing the first chapter. If you are not already there, go to the first chapter by pressing function key **F5** (the Goto key — you'll see a prompt in the status bar in the lower left corner), typing **S3** and pressing the ENTER key. At this point, your insertion pointer should be blinking to the left of "Chapter 1." The status bar at the bottom of your screen should read "Pg 1 Sec 3."

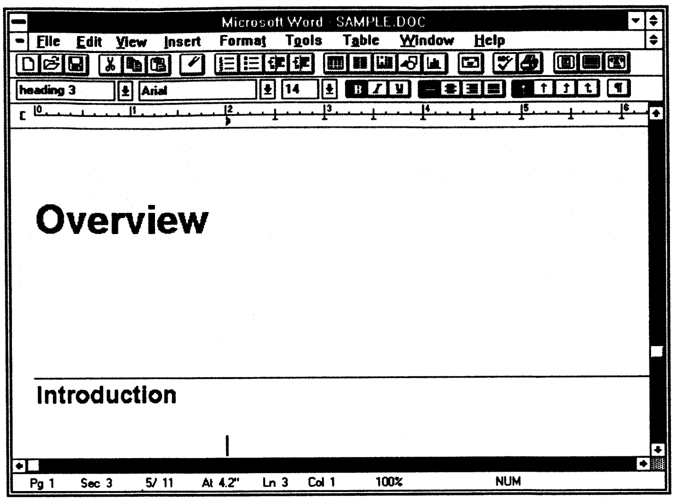
In this next example, you will type the word "Overview" on top of the text place holder. Note that once you change the text of the place holder, the footer for this section will change automatically. (You won't see the footer unless Word is in page layout view.)

To Replace the Place Holder

1. Select the text you want to replace, in this case, **Chapter 1**.
2. Type the text you want to appear as the chapter heading, in this case **Overview,** and press the ENTER key. Notice that the next paragraph has been formatted with a line above it. The Doc-To-Help templates have been designed so that Heading 2 style (which is formatted with a line above) will automatically follow text formatted in Heading 1 style. (For more information on styles see "Applying Styles" on page 24.)

3. Type the word **Introduction** and press the ENTER key. Your screen should look like the one shown below.

*Note: This screen shot was taken while in **Normal** view. Your screen will look different if you are in **Page Layout** view.*



Overview

Introduction

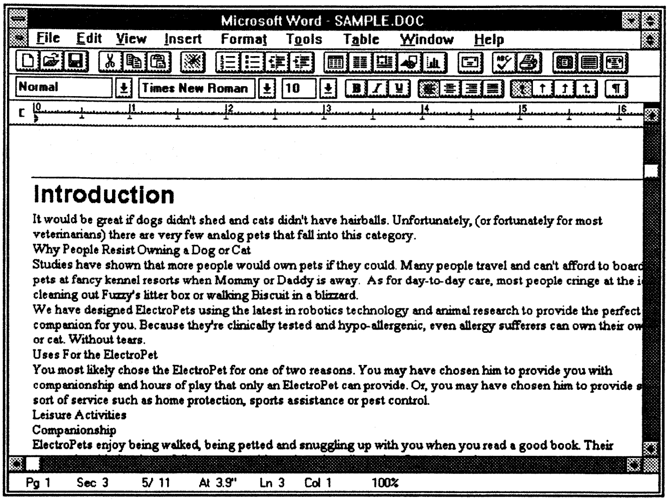
Pg 1 Sec 3 5/ 11 At 4.2" Ln 3 Col 1 100% NUM

Inserting a Practice File

Rather than have you type a lot of text in from scratch, the tutorial includes an unformatted file named PART1.DOC which you can insert into the current document.

To Insert the Practice File

1. The insertion point should be below the word "Introduction."
2. From the **Insert** menu, choose **File**. The Insert File dialog box will appear.
3. Activate the drive and directory where you installed the tutorial files and select **PART1.DOC**. Click **OK**.
4. Press SHIFT+F5 to go back to the previous location in the document. Your screen should look like the one shown on the next page.



Introduction

It would be great if dogs didn't shed and cats didn't have hairballs. Unfortunately, (or fortunately for most veterinarians) there are very few analog pets that fall into this category.

Why People Resist Owning a Dog or Cat

Studies have shown that more people would own pets if they could. Many people travel and can't afford to boar pets at fancy kennel resorts when Mommy or Daddy is away. As for day-to-day care, most people cringe at the i cleaning out Fuzzy's litter box or walking Biscuit in a blizzard.

We have designed ElectroPets using the latest in robotics technology and animal research to provide the perfect companion for you. Because they're clinically tested and hypo-allergenic, even allergy sufferers can own their ow or cat. Without tears.

Uses For the ElectroPet

You most likely chose the ElectroPet for one of two reasons. You may have chosen him to provide you with companionship and hours of play that only an ElectroPet can provide. Or, you may have chosen him to provide s sort of service such as home protection, sports assistance or pest control.

Leisure Activities

Companionship

ElectroPets enjoy being walked, being petted and snuggling up with you when you read a good book. Their

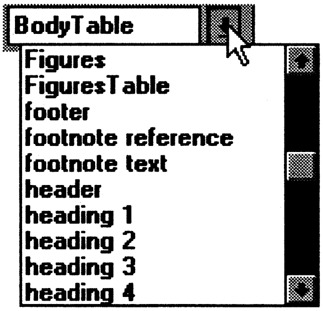
Applying Styles

A style is a pre-defined set of instructions that tells Word how to format a paragraph. By pre-defining styles, you can specify the font, line spacing, indentations, and so on, all in one step.

(See "Doc-To-Help Style Definitions" on page 198 for a listing of all Doc-To-Help styles.)

If the Ribbon is not visible, click the View menu and select Ribbon.

To apply a style, you simply click anywhere in the paragraph you want to format and select the desired style from the style pull-down list box in the Ribbon (see example below). You can also use the Format Style command from the menu.

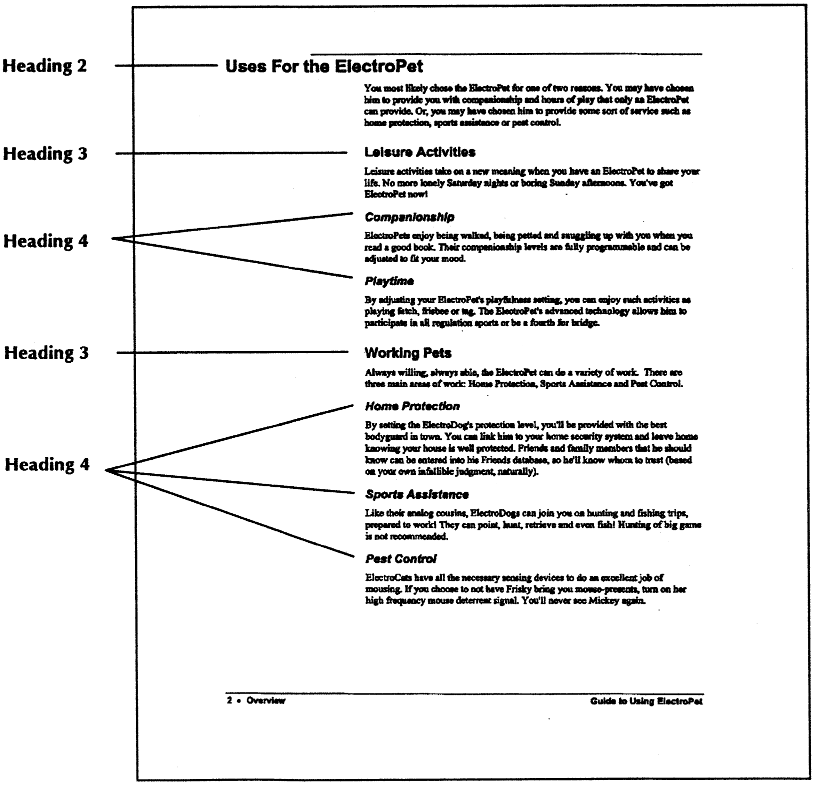


Style list box

Doc-To-Help templates come with a variety of styles to make formatting easy. These styles also play an important role in converting your manual to online help. We've also added shortcut keys to make applying styles faster and easier.

Practice Applying Styles

1. Click anywhere in the paragraph that begins “It would be great if...”
2. Format this paragraph in **Body** style (the shortcut key is CTRL+SHIFT+B).
3. Click anywhere in the line that begins “Why People...”
4. Format this line in **Heading 3** style (CTRL+SHIFT+3).
5. Highlight the next four paragraphs and format them in **Body** style (CTRL+SHIFT+B).
6. Click in the line that begins “Uses for...” and format it in **Heading 2** style (CTRL+SHIFT+2).
7. Finish formatting the section using the styles shown below. (Use Body style except where indicated.)



Uses For the ElectroPet

You most likely chose the ElectroPet for one of two reasons. You may have chosen him to provide you with companionship and hours of play that only an ElectroPet can provide. Or, you may have chosen him to provide some sort of service such as home protection, sports assistance or pest control.

Leisure Activities

Leisure activities take on a new meaning when you have an ElectroPet to share your life. No more lonely Saturday nights or boring Sunday afternoons. You've got ElectroPet now!

Companionship

ElectroPets enjoy being walked, being petted and snuggling up with you when you read a good book. Their companionship levels are fully programmable and can be adjusted to fit your mood.

Playtime

By adjusting your ElectroPet's playfulness setting, you can enjoy such activities as playing fetch, frisbee or tag. The ElectroPet's advanced technology allows him to participate in all regulation sports or be a fourth for bridge.

Working Pets

Always willing, always able, the ElectroPet can do a variety of work. There are three main areas of work: Home Protection, Sports Assistance and Pest Control.

Home Protection

By setting the ElectroDog's protection level, you'll be provided with the best bodyguard in town. You can link him to your home security system and leave home knowing your house is well protected. Friends and family members that he should know can be entered into his Friends database, so he'll know whom to trust (based on your own infallible judgment, naturally).

Sports Assistance

Like their analog cousins, ElectroDogs can join you on hunting and fishing trips, prepared to work! They can point, hunt, retrieve and even fish! Hunting of big game is not recommended.

Pest Control

ElectroCats have all the necessary sensing devices to do an excellent job of mousing. If you choose to not have Frisky bring you mouse-presents, turn on her high frequancy mouse deterrent signal. You'll never see Mickey again.

2 • Overview Guide to Using ElectroPet

Adding a Caption

Anyone who has ever read *Mad* magazine knows that the most interesting information can be found in the margins. Following in this tradition of informational excellence, we have provided a facility that will allow you to place annotations and small illustrations in the left margin of each page.

For the next example you will place a caption (also known as a margin note) in the left margin, next to the paragraph directly underneath the line "Home Protection."

To Insert a Caption

1. Click in the paragraph (or paragraphs) that you want to annotate, in this case the paragraph that begins "By setting the ElectroDog's protection level..."
2. From the **Insert** menu, choose **Caption**. Doc-To-Help will insert a two-column table, and position your cursor in the left column.
3. In the left column, type the caption. For this example type:

   ElectroDogs are compatible with most popular home security systems.

Linking a Caption

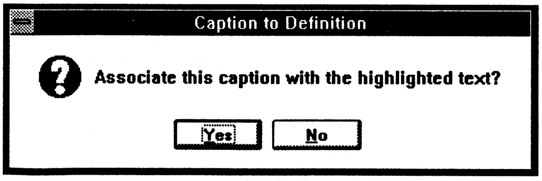
When Doc-To-Help makes your manual into Help, it creates popup **definitions** from captions you link to text in the manual as well as from glossary entries. In the Help screen, text that is linked to a caption will appear with a dotted underline. If you click on the dotted underline text, a definition box will pop up containing the caption the text is linked to.

To Link Text to a Caption

1. With your cursor in the left column, choose **Link to Caption** from the **Insert** menu

If there is a graphic in the right column, Doc-To-Help will give you the option of linking the caption to the graphic.

2. When the dialog box appears, type the text to which the caption should be linked. (This text that will be converted to a definition button in the Help file.) For this example type "home security system".
3. Click **OK.** The text linked to the caption will be highlighted and the following dialog box will appear.



Associate this caption with the highlighted text?

4. Click **Yes.**

Practice

On your own, enter the following caption next to the paragraph after the heading "Pest Control":

> **ElectroCats are available in an industrial strength model for commercial use.**

Link the text "mousing" to this caption.

Writing the Next Chapter

Overview

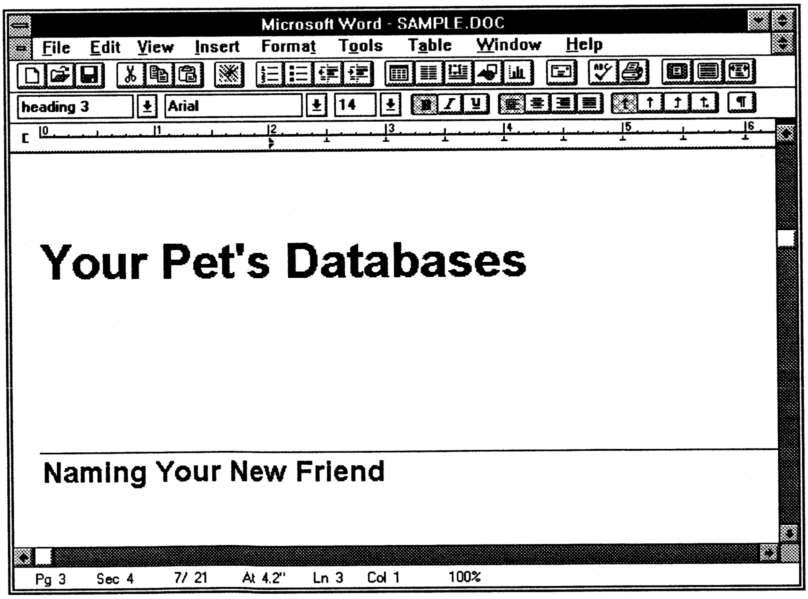
As we saw earlier, new documents have been designed to accommodate two main chapters. If you need to add new chapters (and you probably will) you can do so using the **Insert New Chapter** command.

The Insert New Chapter command behaves differently depending on whether you are working with a single file or multiple files. For more information, see "Insert New Chapter" on page 213.

Writing the Second Chapter

For the next exercise, move to the next section, which you can do easily by pressing function key **F5**, typing **S+1** and pressing **ENTER**. Replace the place holder "Chapter 2" with "Your Pet's Databases" and press the **ENTER** key. Type "Naming Your New Friend" and press the **ENTER** key again. Your screen should look like the one below:

If you are using a non-English version of Word for Windows, typing "S+1" may not work as some versions of Word use a different letter to designate a section.


Microsoft Word - SAMPLE.DOC
File Edit View Insert Format Tools Table Window Help
heading 3
Arial
14
Pg 3 Sec 4 7/ 21 At 4.2" Ln 3 Col 1 100%

Your Pet's Databases

Naming Your New Friend

As you did earlier, instead of typing in the meat of this chapter from scratch, you can insert the practice file PART2.DOC.

To Insert the Second Practice File

1. Your insertion point should be below the text "Naming Your New Friend."
2. Choose **File** from the **Insert** menu.
3. When the dialog box appears, activate the drive and directory that contains the file you want to insert and select **PART2.DOC**.
4. Press **SHIFT+F5** to go back to the previous location in the document. Your insertion point should be at the beginning of the paragraph that starts "ElectroDogs are preprogrammed."

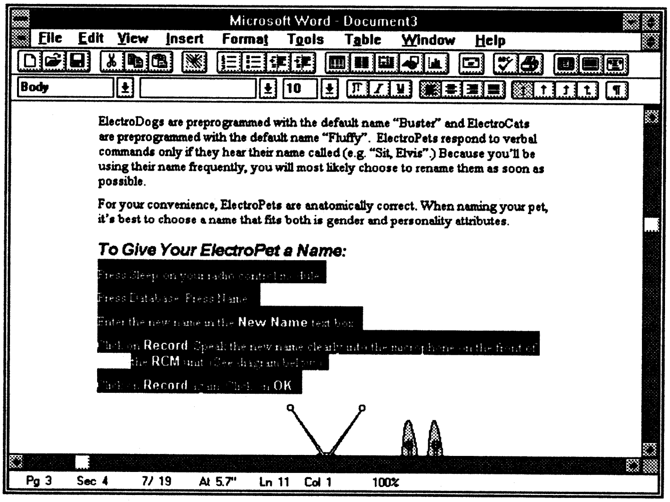
List Style

A pre-defined style called **List** will help you create numbered lists and bulleted lists. List style is indented from the left margin relative to Body style, and contains a hanging indent.

For this next example you will create a numbered list.

To Create a Numbered List

1. Highlight the text you want to format, as in the example shown below.



Microsoft Word - Document3
File Edit View Insert Format Tools Table Window Help
Body
10

ElectroDogs are preprogrammed with the default name "Buster" and ElectroCats are preprogrammed with the default name "Fluffy". ElectroPets respond to verbal commands only if they hear their name called (e.g. "Sit, Elvis".) Because you'll be using their name frequently, you will most likely choose to rename them as soon as possible.

For your convenience, ElectroPets are anatomically correct. When naming your pet, it's best to choose a name that fits both is gender and personality attributes.

To Give Your ElectroPet a Name:

Press Sleep on your radio control module

Press Database. Press Name.

Enter the new name in the **New Name** text box.

Click on **Record**. Speak the new name clearly into the microphone on the front of the **RCM** unit. (See diagram below.)

Click on **Record** again. Click on **OK**.

2. Press **CTRL+SHIFT+L** to apply the List style.
3. Choose **Bullets and Numbering** from the **Tools** menu. The Bullets and Numbering dialog box will appear.
4. Click the **Numbered List** option button. Choose the desired numbering scheme from the **Number** group box. Click **OK**.

Special Bold

You may have noticed that every time a new term is introduced and whenever a user is asked to choose a particular option, the term and option are formatted in the same font used for headings. You can apply this same formatting using the **Format Special Bold** command.

To Apply Special Bold

Special Bold is a toggle. Applying it a second time turns special formatting off.

1. Highlight the word or phrase you want to format, in this case the word **Sleep** in the first item of the numbered list.
2. From the **Format** menu, choose **Special Bold**, or press CTRL+SHIFT+S.
3. Apply Special Bold to the words **Database** and **Name**.

Taking Screen Shots

If a dialog box is visible, only the dialog box is copied to the clipboard. To copy the entire display, press PRINT SCRN *by itself.*

Many people don't know it, but Windows comes with a built-in screen capture capability. Simply press ALT+PRINT SCRN and a picture of the active window is copied to the clipboard. If you don't need to take a lot of screen shots, this built-in facility will be more than adequate.

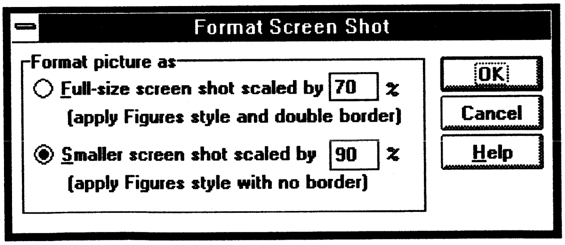
Formatting Screen Shots

Once you have taken a screen shot and pasted it into a Doc-To-Help document, you will need to format it so that it is scaled properly. You can do this easily using the **Format Screen Shot** command.

To Format a Screen Shot of a Dialog Box

1. Click on the screen shot you want to format, in this case the Remote Control Module diagram that follows list item 5. You can tell if the graphic is selected if small boxes surround the graphic.
2. From the **Format** menu, choose **Screen Shot**. The following dialog box will appear:

"Full-size" is for formatting screen shots of the entire screen. "Smaller" is for formatting dialog boxes, message boxes, etc.



Format Screen Shot

Format picture as

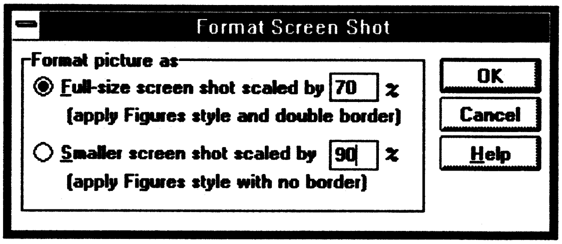
○ Full-size screen shot scaled by 70 %
(apply Figures style and double border)

● Smaller screen shot scaled by 90 %
(apply Figures style with no border)

3. Select the **Smaller screen shot** option.
4. Click **OK**. Doc-To-Help will scale the picture and apply Figures style formatting.

To Format a Full-size Screen Shot

1. Click on the screen shot you want to format. For this example, select the full screen shot of the Friends Database dialog box which is on the next page of the document you are editing. You can tell if the graphic is selected if small boxes surround the graphic.
2. From the **Format** menu, choose **Screen Shot**. The following dialog box will appear:



Format picture as

● Full-size screen shot scaled by 70 %
(apply Figures style and double border)

○ Smaller screen shot scaled by 90 %
(apply Figures style with no border)

3. Select **Full-size** and click **OK**. Doc-To-Help will scale the graphic, add a double border and apply Figures style formatting.

Bonus Exercise

Replace the empty box on the cover page with the picture file "PETS.WMF."

1. Press CTRL+HOME to move to the top of the document.
2. Click on the empty frame on the title page.
3. Press DELETE. The empty frame will disappear.
4. From the **Insert** menu choose **Picture**.
5. From the dialog box choose "PETS.WMF." It is located in the tutorial directory.
6. Format the picture using the **Format Screen Shot** command.

Conserving File Size with Monochrome Screen Shots

Screen shots — especially color screen shots — take up a lot of disk space and will increase the size of the Help file. You can conserve a tremendous amount of disk real estate by taking monochrome screen shots. You can easily take monochrome screen shots by changing your Windows system settings to monochrome VGA and changing your Control Panel colors to Monochrome.

You can also reduce the size of your documents (as well as increase the speed of scrolling and printing) by using a utility from WexTech Systems called Quicture. Quicture is a Word for Windows utility that allows you to offload graphics to disk, replacing each graphic with a descriptive placeholder. In the process of offloading, Quicture also compresses the graphic. Your documents scroll quickly, save quickly, print quickly and take up much less space on your hard drive. You can print draft copies with the pagination and layout of your document intact. You can view or print any of your graphics at any time and easily print the complete document whenever you want. (For more information on Quicture, contact WexTech Systems.)

Inserting a Cross Reference

Throughout your manual you may want to place references to information on other pages such as

> **For more information on Hairballs, see "Health Needs" on page 203.**

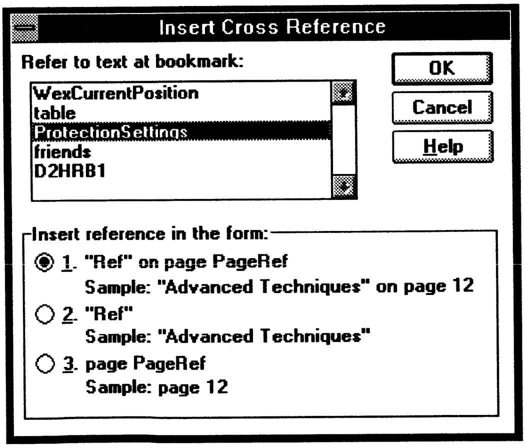
Doc-To-Help will convert cross references to hypertext jumps in your Help file.

Creating a cross reference is usually a two-step process. The first step is to insert a bookmark at the location you want to reference. The second step is to insert a cross reference that refers to the bookmarked text.

For this example, we've already done the first step. We've created a bookmark which covers the heading "Protection Settings" further on in the ElectroPet manual.

To Insert a Cross Reference

1. Position your cursor where you want the reference to appear. In this example, position your insertion point at the end of the first paragraph in the "Friends Database" section which ends "...people your ElectroDog should recognize." Type **For more information, see**
2. From the **Insert** menu choose **Cross Reference (X-Ref)**. The **Insert Cross Reference** dialog box will appear as shown below.



Insert Cross Reference

Refer to text at bookmark:

WexCurrentPosition
table
ProtectionSettings
friends
D2HRB1

Insert reference in the form:

● 1. "Ref" on page PageRef
Sample: "Advanced Techniques" on page 12

○ 2. "Ref"
Sample: "Advanced Techniques"

○ 3. page PageRef
Sample: page 12

3. Select the bookmark and the form of the reference you want to insert. In this case choose "ProtectionSettings" and leave **Insert reference in the form:** at "1."

4. Click **OK**. Doc-To-Help will insert a cross reference/page reference like the one shown below.

   For more information, see "Protection Settings" on page 9.

5. Type a period (.).

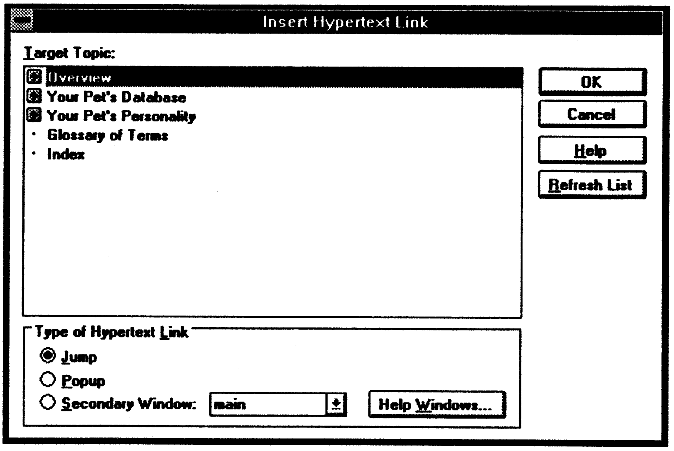
Inserting a Hypertext Link

There will probably be places where you want a cross reference jump to appear in the help file but don't want it to appear in the manual. You can create these "help only" jumps using the Insert Hypertext Link command.

For this next example, let's suppose you want to make it so that clicking on the words "Friends Database" in the help file will jump to the section on the Friends Database.

To Insert a Hypertext Link

1. Highlight the text "companionship setting" (it's in the same paragraph as the cross reference you created earlier).
2. From the **Insert** menu, choose **Hypertext Link**. Doc-To-Help will display the message asking you if you want to refresh the list of topics to which you can jump.
3. Click **Yes**. Doc-To-Help will assemble a list of topics based on the headings in your document, and display a dialog box similar to the one shown below.



Insert Hypertext Link

Target Topic:

- Overview
- Your Pet's Database
- Your Pet's Personality
- Glossary of Terms
- Index

Type of Hypertext Link

● Jump

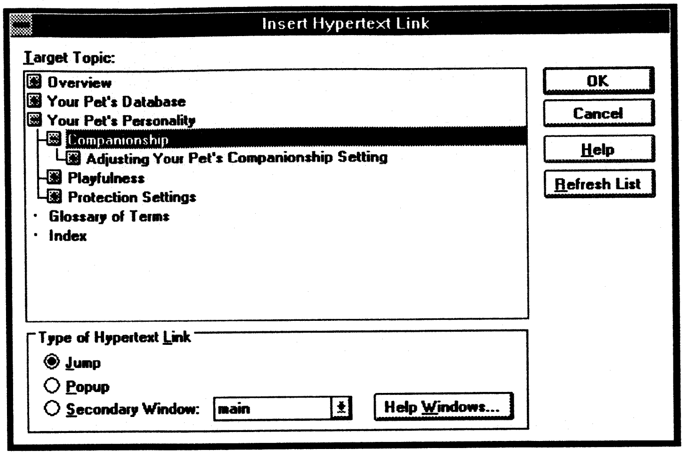
○ Popup

○ Secondary Window: main

Insert Hypertext Link dialog box showing heading 1 topics only

4. Double-click on **Your Pet's Personality** to see that topic's sub-topics. (If you are using the keyboard, highlight **Your Pet's Personality** and press the plus sign (+).)

5. With the "Your Pet's Personality" topic partially expanded, double-click on Companionship. Your screen should look like the one shown below.



Insert Hypertext Link

Target Topic:

- Overview
- Your Pet's Database
- Your Pet's Personality
  - Companionship
    - Adjusting Your Pet's Companionship Setting
  - Playfulness
  - Protection Settings
- Glossary of Terms
- Index

Type of Hypertext Link

● Jump
○ Popup
○ Secondary Window: main

For more information on Relate fields, see "Relate Field Syntax" on page 179.

6. Select **Adjusting Your Pet's Companionship Setting** and click **OK**. Doc-To-Help will insert a special Relate field which will be turned into a jump when you make the manual into help. If you want, you can examine this field by turning field codes on.

Before continuing, turn field codes off.

Inserting and Formatting a Table

The **Table Standard Table** command will enable you to both insert and format tables more easily than with the regular Word for Windows table commands. The primary advantages of using the Doc-To-Help table command instead of the regular Word table commands are:

- Tables will be indented to line up with body text.
- Cells will be pre-formatted in TableHeading and TableText styles.
- Borders can be applied automatically.

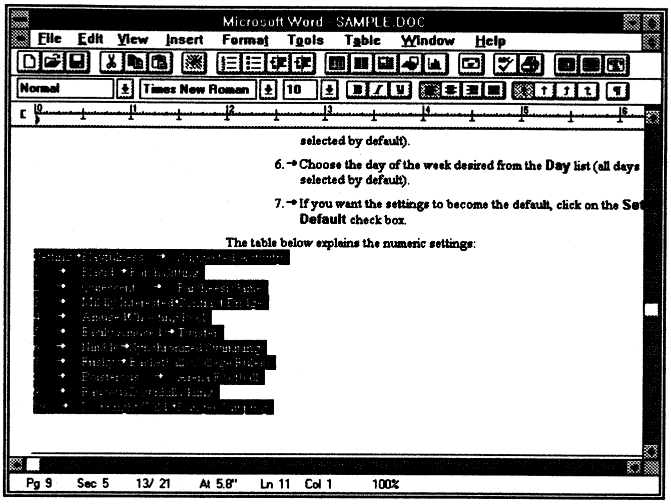
The Standard Table command will behave differently depending on whether your cursor is in a table and whether any text is selected.

- If the cursor is inside a table, the command will help you format the table.
- If the cursor is not inside a table and text is selected, the command will help you convert that text into a table and apply formatting.
- If the cursor is not inside a table and no text is selected, the command will present a dialog box asking you how large the table should be.

For the next example, let's format the playfulness chart which is formatted with tabs instead of a table. To move to this table press function key **F5**, and type **Table**. Press **Enter** and the insertion point will be placed above the table.

To Turn Text into a Standard Table

1. Highlight the text you want to format, as shown below.



Microsoft Word - SAMPLE.DOC
File Edit View Insert Format Tools Table Window Help
Normal
Times New Roman
10

selected by default).

6.→Choose the day of the week desired from the **Day** list (all days selected by default).

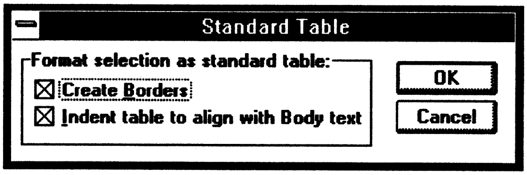
7.→If you want the settings to become the default, click on the **Set Default** check box.

The table below explains the numeric settings:

Setting → Playfulness → Suggested Activity
1 → Bored → Porch Sitting
2 → Quiescent → Pointless Game
3 → Mildly Interested → Contract Bridge
4 → Amused → Shooting Pool
5 → Easily Amused → Twister
6 → Nimble → Synchronized Swimming
7 → Frisky → Basketball/College Rules
8 → Boisterous → Arena Football
9 → [illegible] → Downhill Skiing
10 → [illegible] → [illegible]

Pg 9 Sec 5 13/ 21 At 5.8" Ln 11 Col 1 100%

2. From the **Table** menu, choose **Standard Table**. The following dialog box will appear:



Standard Table
Format selection as standard table:
☒ Create Borders
☒ Indent table to align with Body text
OK
Cancel

3. Click **OK**.

For more information on using this feature, see "Table Menu" on page 242 and "Tables and Borders" on page 109.

Paragraph Spacing

You may have noticed that the line spacing between the line before the table and the table itself is a bit on the tight side. You can rectify this problem using the **Format Adjust Spacing** command.

The Adjust Spacing command will allow you to control the spacing before and after a paragraph, one point at a time.

To Add More Space After a Paragraph

1. Click in the paragraph you want to have space after, in this case the one that begins "The table below explains the numeric settings:..."
2. Press CTRL and the numeric keypad plus sign (+).
3. Repeat step two until you are satisfied with the line spacing.

For a list of all the line spacing shortcut keys, see "Shortcut Keys" on page 203.

Glossary of Terms

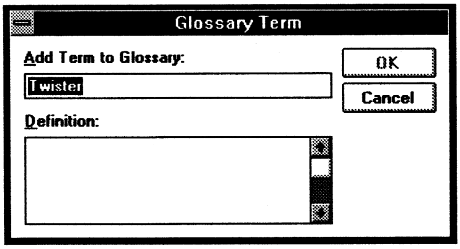
As you assemble your documentation, you will undoubtedly introduce concepts and terms which are new to your reader. Whenever you introduce a new term or concept, you should add that item to the Glossary of Terms using the **Insert Glossary Term** command.

The Glossary of Terms is also used to create popup definitions in the Help file. (See "Definitions" on page 49.)

For this next exercise, let's suppose you want to add the terms "Twister" and "Arena Football" to the Glossary.

To Add a Term to the Glossary

1. Highlight the term or phrase you want to add, in this case the word "Twister" which can be found in the middle of the table.
2. From the **Insert** menu, choose **Glossary Term**. The following dialog box will appear:



Add Term to Glossary: Twister

Definition:

3. You can modify the glossary term in the Add Term to Glossary text box so it looks the way you'd like it to appear in the glossary. Since "Twister" is already capitalized, you can leave it as is.
4. Press TAB. Type the definition of the glossary term in the Definition box. The definition for Twister is:

   Popular '60s party game where participants place available body parts on game board with brightly colored circles.
5. Click **OK**.

On your own, enter a glossary term for Arena Football.

Basic Indexing

Overview

If you're anxious to see what the manual looks like, and how to turn the manual into Help, you can skip this section and learn about indexing later.

While indexing is about as much fun as root canal, it is definitely worth the effort. The index is the first place your readers will look to find information on a particular subject. The better your index, the easier it will be to find that information. In addition, your document's index will seed the online help keyword search list — and the keyword search is probably the most often used feature of online help.

The Doc-To-Help indexing commands will not only allow you to produce a complete and thorough index that even the most persnickety of reviewers will laud, it will also allow you to create that index much faster than if you had used the standard Word commands.

Creating an index is a three step process:

1. Add all the words you want to the Index Target List. The index target list contains all the words (index targets) that you may want to index. You do this with the **Add to Index Target List** command.
2. Enter all the index tags you want to associate with each of the index targets in the index target list. An **Index Tag** is the actual term or phrase that will appear in the index.
3. Generate the formatted index using the **Tools Indexing Build Index** command.

For a complete discussion of commands, see "Indexing" on page 65.

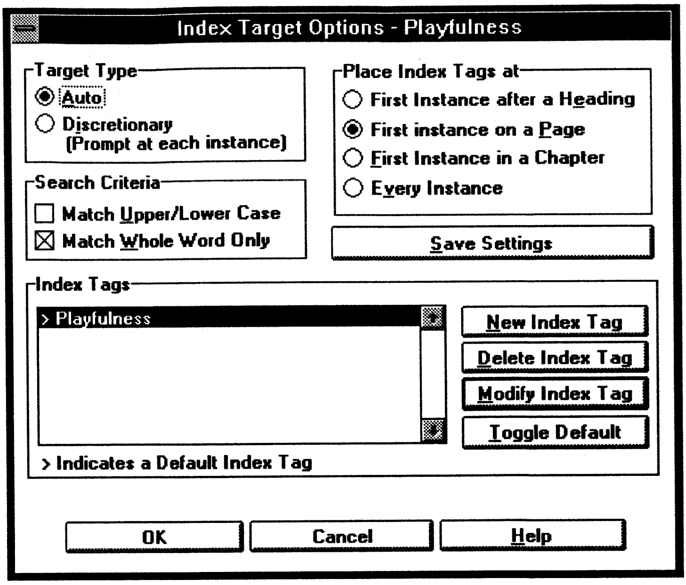
For this next series of examples, let's suppose you want to add "playfulness" and "protection" to the index list. If you've skipped this section the first time you did the tutorial, use **Edit Find** to search for "playfulness."

To Add an Index Target to the Index Target List

1. Highlight the word or phrase you want to add, in this case **"Playfulness"**.
2. From the **Tools** menu, choose **Indexing....**
3. When the dialog box appears, choose **Add to Index Target List**.

4. The **Index Target Options** dialog box will appear as shown below.

If the Index Target Options dialog box does not appear, click on Tools, choose Doc-To-Help options and make sure Prompt on Add to Target List is selected.



Index Target Options - Playfulness

Target Type
● Auto
○ Discretionary
(Prompt at each instance)

Place Index Tags at
○ First Instance after a Heading
● First instance on a Page
○ First Instance in a Chapter
○ Every Instance

Search Criteria
□ Match Upper/Lower Case
☒ Match Whole Word Only

Index Tags
> Playfulness

> Indicates a Default Index Tag

Multiple index tags are necessary for a truly useful index. For example, it is unlikely that all of your readers will look up "Playfulness" under the index tag "Playfulness." They may first look for "Personality Attributes" or perhaps make a beeline for "Games." Both of these entries, along with "Playfulness," should be in your index. You can assign multiple index tags using the **New Index Tag** button.

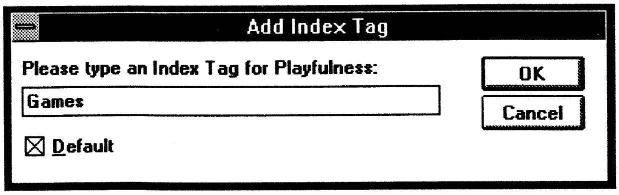
Auto indexing is the default Target Type.

As the entry currently stands, Doc-To-Help will search for the first instance on a page for the text "Playfulness" and insert the index tag "Playfulness." It will do this automatically, as long as the **Target Type** is set to **Auto**. For a discussion of **Discretionary Indexing** see "Discretionary Indexing" on page 69.

We will add two more index tags.

5. Click **New Index Tag**. When the dialog box appears, type "**Games**" and make sure **Default** is selected, as shown below.

We'll explain what "Default" means in the chapter devoted to indexing.



Add Index Tag

Please type an Index Tag for Playfulness:
Games

☒ Default

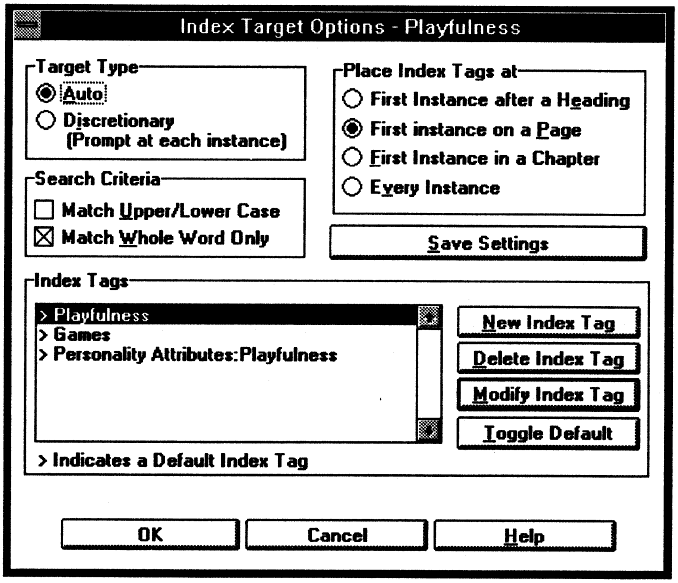
6. Click **OK** then click **New Index Tag** again.

7. When the dialog box appears, type **Personality Attributes:Playfulness**.

   Entering a colon between words will create a multiple-level index like the one shown below:

   Personality Attributes
   Companionship 5, 6, 9, 11, 13
   Playfulness 7, 9
   Protection 2, 8, 11, 13

8. Make sure **Default** is selected and click **OK**. The dialog box should look like the one shown below.



Index Target Options - Playfulness

Target Type
● Auto
○ Discretionary (Prompt at each instance)

Search Criteria
□ Match Upper/Lower Case
☒ Match Whole Word Only

Place Index Tags at
○ First Instance after a Heading
● First instance on a Page
○ First Instance in a Chapter
○ Every Instance

Index Tags
> Playfulness
> Games
> Personality Attributes:Playfulness

> Indicates a Default Index Tag

These settings indicate that when you build the index Doc-To-Help will search for the first occurrence on each page of the term "Playfulness" and insert the index tags "Playfulness," "Personality Attributes:Playfulness" and "Games."

9. Make sure **First Instance on a Page** is selected in the **Place Index Tags at** group box and that **Match Whole Word Only** is the only check box selected.
10. Click **OK.**
11. Highlight the word "**Protection**" (found below the table).
12. This time you'll use the shortcut keys instead of the menu. Press CTRL+SHIFT+A.
13. The **Index Target Options** dialog box will appear. Click **New Index Tag**.
14. When the Add Index Tag dialog box appears, type **Personality Attributes:Protection** and click **OK**.
15. When the Index Target Options dialog box reappears, click **OK**.

Building the Index

Once you have assembled your index target list and assigned your index tags you are ready to build the index.

Since we instructed Doc-To-Help not to prompt us as it enters index tags into the document, Doc-To-Help will automatically assign default index tags associated with each index target.

To Build the Index

1. From the **Tools** menu, choose **Indexing**. From the Indexing dialog box choose **Build Index**.
2. A message box will appear asking if you've set up the Index List the way you want it. Click **Yes**.

Doc-To-Help will find the first instance on each page of every index target, and insert the appropriate index tag. The index (at the very end of the document) will still display "Error! No index entries found."

Printing

Updating Fields

Word for Windows may be set up to update all the fields in your document whenever you print. These include the Table of Contents field, the Index field and any other fields (such as Ref and Pageref). You can change this automatic updating of fields by choosing File Print Options and turning Update Fields off.

Make sure Field Codes and Hidden text are turned off when you update the fields. Otherwise, pagination will not be correct and you won't be able to view the results of the Update Fields command. You can access both of these setting through the Tools Options View dialog box.

To manually update the fields, select any text produced by the field and press function key **F9**. (For example, move to the Table of Contents section, highlight part of the Table of Contents, and press **F9**.) Please note that if you have the entire document selected when you update the fields, Doc-To-Help gives you the option of assembling a list of headings that can be accessed by the Insert Hypertext Link command. (See "Insert Hypertext Link" on page 210.)

Sorting the Glossary of Terms

You may want to sort the glossary of terms before printing your document. You can do this by choosing **Sort Glossary of Terms** from the **Tools** menu.

Review

What You've Done So Far

In a moment, we'll convert the ElectroPet manual into online help. Before we do that, let's take a moment to review what we've learned so far. We've seen how to:

- Use **File New** to create a document based on one of the Doc-To-Help templates.
- Set the fonts for the document.
- Apply styles to format the document.
- Insert and Link Captions.
- Create lists by applying the List style and using Bullets and Numbering.
- Emphasize new terms with Special Bold.
- Format Screen Shots.
- Insert Cross References.
- Insert Hypertext Links
- Insert and format a table.
- Create and edit the Index Target List.
- Build an index.
- Create a glossary of terms.
- Sort the glossary entries.
- Modify space after paragraphs using **CTRL + PLUS SIGN**.
- Save and print the manual.

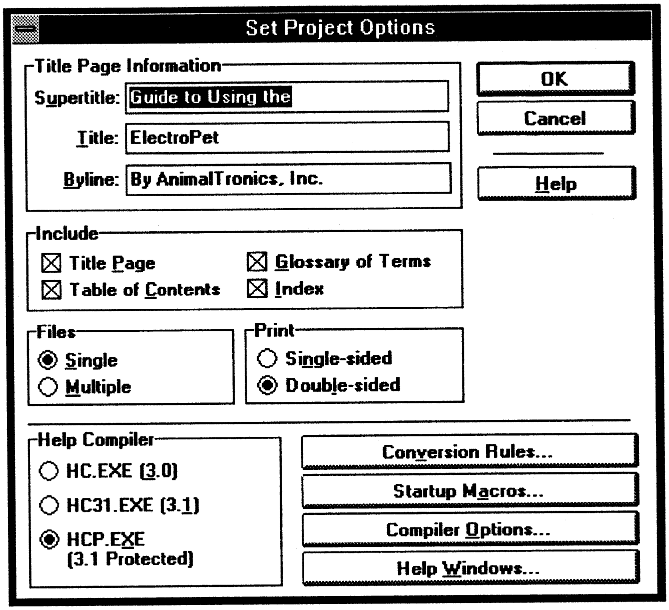
To see more detailed information about each topic, refer to the "Reference" which begins on page 195.

Making the Manual into Help

Before making the manual into a help file, we should set which help compiler we will be using as well as set the size, placement and color for the help window that will display the help file.

To Set the Help Options

1. From the **Format** menu choose **Set Project Options**. When the Set Project Options dialog box appears, click **Help Options>>**. The Set Project Options dialog box will expand, as shown below.



Set Project Options

Title Page Information
Supertitle: Guide to Using the
Title: ElectroPet
Byline: By AnimalTronics, Inc.

Include
☒ Title Page ☒ Glossary of Terms
☒ Table of Contents ☒ Index

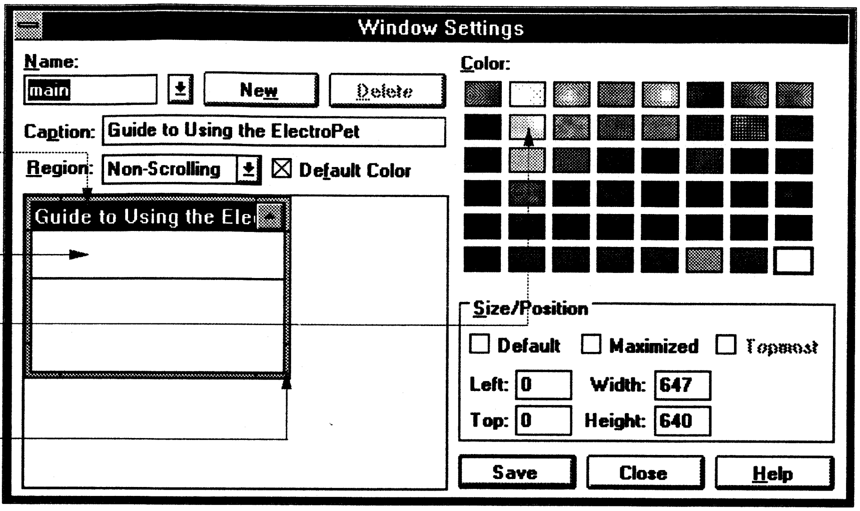
Files
● Single
○ Multiple

Print
○ Single-sided
● Double-sided

Help Compiler
○ HC.EXE (3.0)
○ HC31.EXE (3.1)
● HCP.EXE (3.1 Protected)

HCP will take advantage of your system's extended memory; the other compilers will not.

2. Choose the compiler you want to use (in this example HCP—the Protected-Mode Help compiler for Windows 3.1).
3. Click **Help Windows**. The Window Settings dialog box will appear, as shown below.



Click on the title and drag to move the help window.

Click here to select the non-scrolling region.

Click here to select a color for the selected region.

Click here and drag the border to change the size of the window.

Name: main

Caption: Guide to Using the ElectroPet

Region: Non-Scrolling ☒ Default Color

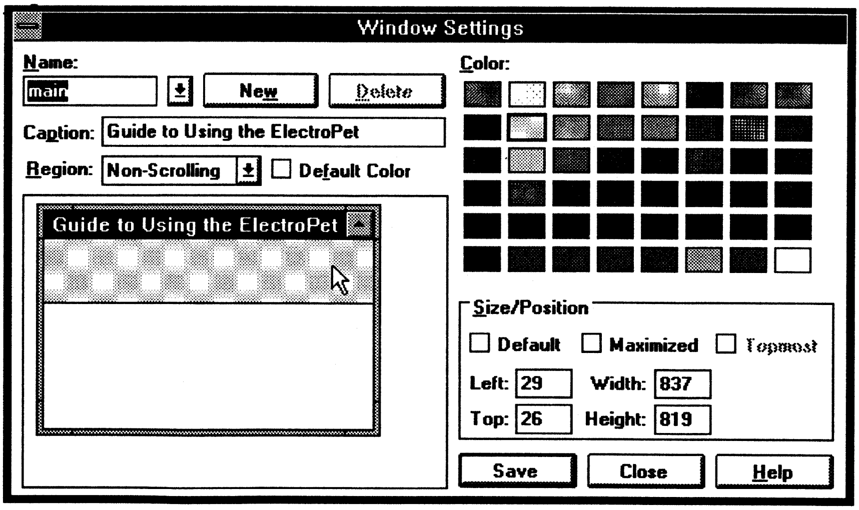
□ Default □ Maximized □ Topmost

Left: 0 Width: 647

Top: 0 Height: 640

The Window Settings dialog box showing the size and shape of the ElectroPet help window relative to a full-sized screen.

4. Click in the "Guide to Using the..." title bar and drag the window down and to the right a little bit.
5. Click on the bottom right border of the "Guide" window and make the window larger by dragging down and to the right.
6. Click in the non-scrolling region, and change the color to pale yellow. Your screen should look like the one shown below.



Name: main

Caption: Guide to Using the ElectroPet

Region: Non-Scrolling □ Default Color

□ Default □ Maximized □ Topmost

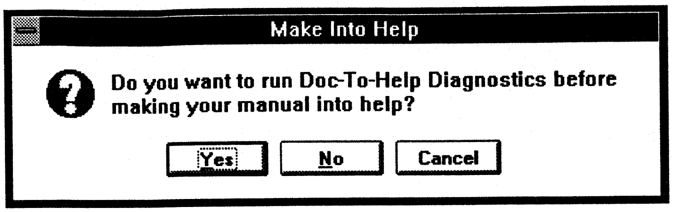
Left: 29 Width: 837

Top: 26 Height: 819

7. Click **Save**, then click **Close**.
8. When the Set Project Options dialog box reappears, click **OK**.

To Make the Manual Into Help

1. From the **Format** menu, choose **Make Into Help**. The following message box will appear.



Make Into Help

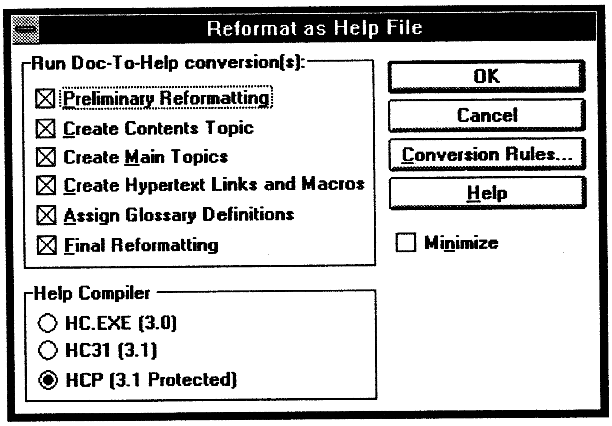
Do you want to run Doc-To-Help Diagnostics before making your manual into help?

2. Since we won't run Doc-To-Help Diagnostics now, click **No**. (See "Tools Doc-To-Help Diagnostics" on page 236 for more information.)

   Doc-To-Help will save and close your file in the standard Word format (.DOC) and then save the file as an RTF file. After performing a little additional house-cleaning, Doc-To-Help will present the following dialog box:

If you will be making changes to the manual that you want to see in the Help file, edit the .DOC file as desired, then choose Make Into Help again.

If you click Cancel, you can access this dialog box again by choosing ***Reformat as Help*** *from the* ***Format*** *menu.*



Reformat as Help File

Run Doc-To-Help conversion(s):

☒ Preliminary Reformatting
☒ Create Contents Topic
☒ Create Main Topics
☒ Create Hypertext Links and Macros
☒ Assign Glossary Definitions
☒ Final Reformatting

□ Minimize

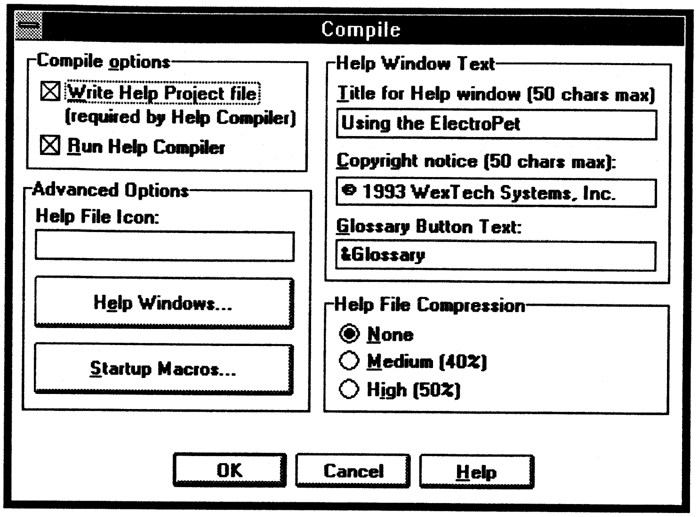
Help Compiler

○ HC.EXE (3.0)
○ HC31 (3.1)
● HCP (3.1 Protected)

3. With all the options selected, Click **OK**. (For information on reformatting options, see "Format Reformat as Help" on page 227.)

4. When Doc-To-Help asks if you want to compile the reformatted file into Help, click **Yes**. A dialog box similar to the following will appear:

The dialog box options will be different if you use the Windows 3.0 Help compiler (HC.EXE).You select which compiler to use by clicking the ***Help Options*** *button from the* ***Set Project Options*** *dialog box.*



Compile options

☒ Write Help Project file (required by Help Compiler)

☒ Run Help Compiler

Advanced Options

Help File Icon:

Help Window Text

Title for Help window (50 chars max): Using the ElectroPet

Copyright notice (50 chars max): © 1993 WexTech Systems, Inc.

Glossary Button Text: &Glossary

Help File Compression

● None

○ Medium (40%)

○ High (50%)

Windows 3.1 Protected Mode Compiler Selected

5. If you want, edit the Help window title or the copyright notice and click **OK**. Doc-To-Help will write the Help Project (HPJ) file to your project directory. Then it will "shell" to DOS and compile your HPJ file into an HLP file. You should see dots appear on the screen as the file is compiled. When compilation is complete, Doc-To-Help will reactivate Word.
6. Choose **Run Help** from the **Tools** menu to check the file. (See "Appendix: Accessing Help" on page 245 for suggested ways to use the Help file.)

Important Note: *If you choose to edit the RTF file on your own, when you go to save the file, Word may ask you if you want to save the file in a native Word for Windows format. Do NOT take Word up on this offer. The file must be saved in an RTF format.*

Stuff You Can Delete

Unless you're a hacker who likes playing with RTF files and bitmaps (or if you will be making changes to your source documentation and recompiling), you can delete many of the files Doc-To-Help leaves behind after you have converted your document into Help. These "no longer needed files" are:

- the RTF file (which will be re-created if you convert your manual into Help again),
- the Help project file (HPJ file) that Doc-To-Help writes before it hands off the RTF file to the Help Compiler,
- D2H_COMP.BAT (it it exists),
- DOC2HELP.INF (if it exists),
- D2H_HEAD.BIN (if it exists),
- the ERR file containing any errors encountered when creating the HPJ file, and
- all the bitmap files and Windows metafiles placed in the BITMAP directory. Like the RTF file, these files will be re-created if you convert your manual into Help again.

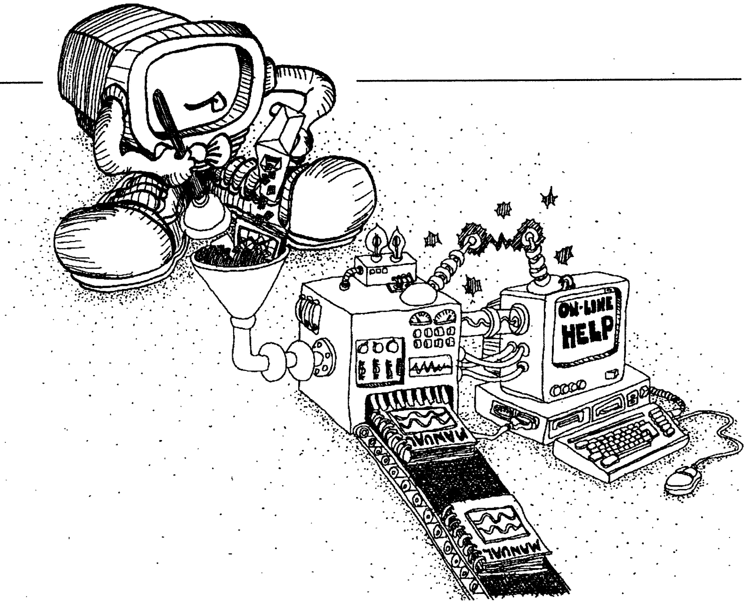
If you will be compiling the file again, and need to preserve context string mappings, make sure that you do not *delete the HPJ file.*

Performance Tips

You can increase the speed at which Doc-To-Help converts your manuals into help by doing any and all of the following:

- Before making the manual into help, choose Draft from the View menu.
- Make sure your document "fits" within your screen; that is, make sure that you can read all the text on your screen and that no text goes beyond the right edge of the screen. To make the text fit, choose Zoom from the View menu and change the magnification as needed.
- Turn the Minimize check box on in the Reformat as Help dialog box.

How Doc-To-Help Turns Manuals Into Online Help



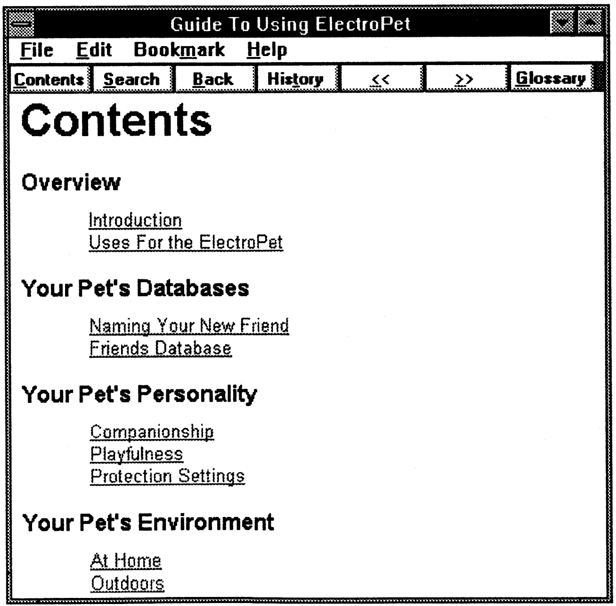
How Doc-To-Help Turns Manuals into Online Help

Components of Online Help

Topics

The figure below shows the contents topic screen of a sample Help because it was derived directly from the table of contents. In this example, anything formatted in Heading 1 style appears as static text on the contents topic while anything formatted in Heading 2 style is accessible as a "jump." In the example below, we can glean that "Overview" and "Your Pet's Databases" have been formatted in Heading 1 style, while "Introduction," "Uses For the ElectroPet," "Naming Your New Friend" and "Friends Database," and so on, have been formatted in Heading 2 style.

Clicking on an underlined item will make the Help system jump to the help topic about that item.



Contents

Overview

Introduction
Uses For the ElectroPet

Your Pet's Databases

Naming Your New Friend
Friends Database

Your Pet's Personality

Companionship
Playfulness
Protection Settings

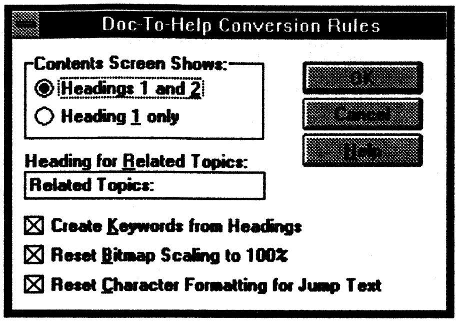
Your Pet's Environment

At Home
Outdoors

Contents Topic showing heading 1 and 2 text

You can access the Conversion Rules dialog box from either the Format Set Project Options dialog box or from the Format Reformat as Help dialog box.

You determine how much detail appears on the contents topic through Doc-To-Help's Conversion Rules dialog box.

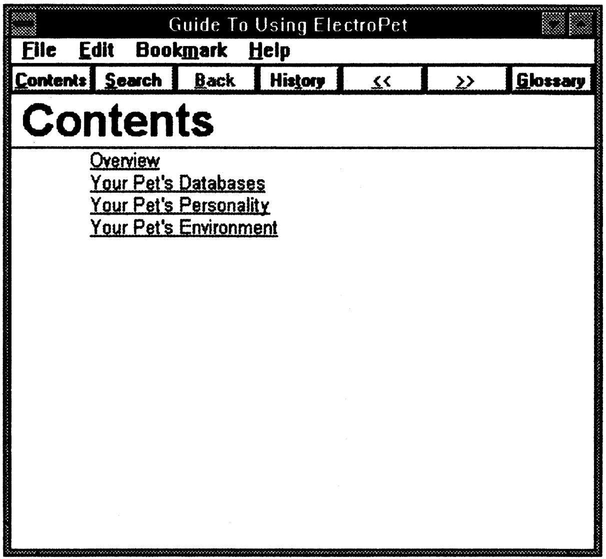


Contents Screen Shows:
● Headings 1 and 2
○ Heading 1 only

Heading for Related Topics:
Related Topics:

☒ Create Keywords from Headings
☒ Reset Bitmap Scaling to 100%
☒ Reset Character Formatting for Jump Text

By default, single file project's will show Heading 1 and 2 text on the contents topic. If you were to change the Contents Screen Shows option to Heading 1 only, the resulting help file would look like the one shown below.

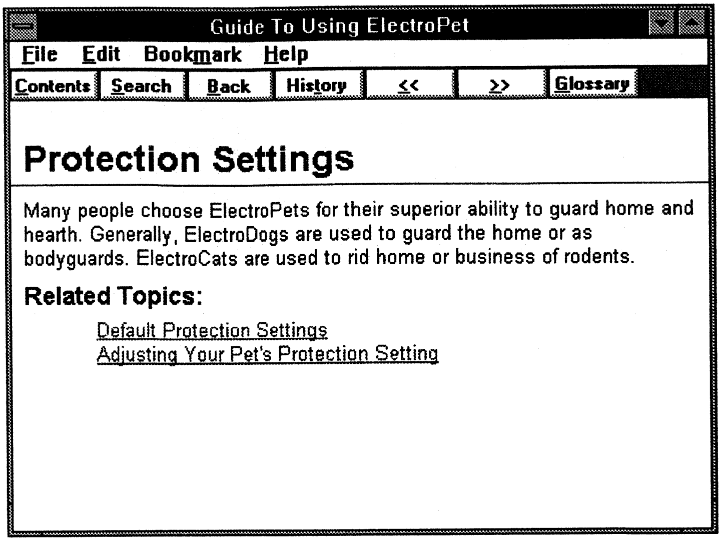


Contents

Overview
Your Pet's Databases
Your Pet's Personality
Your Pet's Environment

Contents Topic showing Heading 1 only

For this next example, let's suppose we click on "Protection Settings." Clicking on this item will present a screen like the one shown below.


Guide To Using ElectroPet
File Edit Bookmark Help
Contents Search Back History << >> Glossary

Protection Settings

Many people choose ElectroPets for their superior ability to guard home and hearth. Generally, ElectroDogs are used to guard the home or as bodyguards. ElectroCats are used to rid home or business of rodents.

Related Topics:

Default Protection Settings
Adjusting Your Pet's Protection Setting

Note: Additional hypertext jumps can be placed anywhere in the Help file using the ***Insert Hypertext Link*** *menu command.*

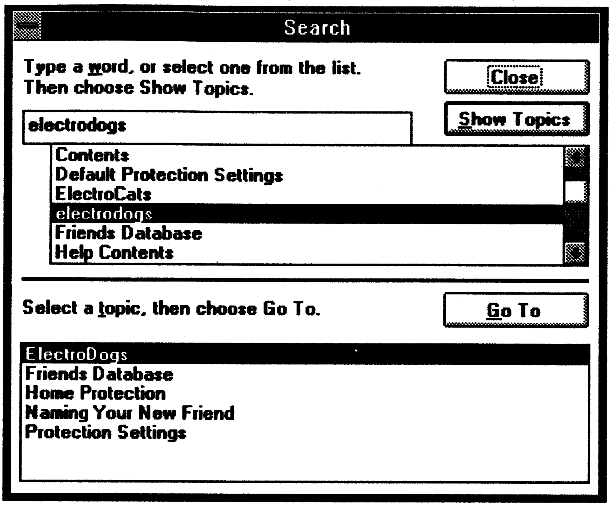
Anything formatted in a Heading 2, Heading 3 or Heading 4 style becomes a related topic of whatever main topic immediately precedes it. (You have the option to make paragraphs formatted in Heading 2 style into jumps from the contents page or related topics of the preceding Heading 1 topic.) In this example, "Protection" is formatted in the Heading 2 style and "Default Protection Settings" and "Adjusting Your Pet's Protection Setting" are formatted in the Heading 3 style.

Definitions

You may also notice that some words and phrases are formatted with a dotted underline. When the user clicks on text formatted in this fashion, a definition or clarification about that underlined text will appear in a popup window. These definitions are created by Doc-To-Help based on the captions placed in the left margins.

Keywords and Search

One of the most useful features of the Windows Help engine is its ability to jump to topics based on a keyword search. Consider the example shown below.



Type a word, or select one from the list.
Then choose Show Topics.

electrodogs

Contents
Default Protection Settings
ElectroCats
electrodogs
Friends Database
Help Contents

Select a topic, then choose Go To.

ElectroDogs
Friends Database
Home Protection
Naming Your New Friend
Protection Settings

In this example, we've chosen **Search** from the **Help** menu, selected **ElectroDogs** from the list and clicked on the **Show Topics** button. The Help application has displayed a list of available topics associated with the keyword "**ElectroDogs**."

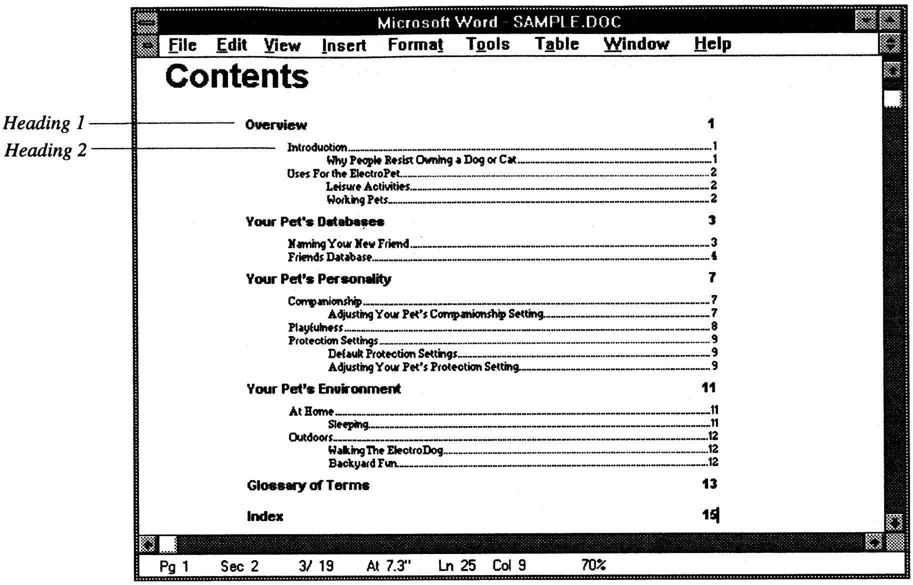
The keywords list at the top of the dialog box is derived from text formatted in the heading styles in the original document and index tags placed in the original document. The topics listed in the bottom of the document are topics which contain the selected keyword. In the original document that generated this example, Doc-To-Help found index tags with the word "**ElectroDogs**" in the text following the headings **"ElectroDogs," "Friends Database," "Naming Your New Friend," "Protection"** and **"Sports Assistance."**

A good index produces a particularly good set of keywords, so it's well worth being meticulous about producing a good index.

How Doc-To-Help Does It

Table of Contents Becomes the Main Help Screen

If you have a table of contents that looks like this...


Microsoft Word - SAMPLE.DOC
File Edit View Insert Format Tools Table Window Help
Heading 1
Heading 2
Pg 1 Sec 2 3/ 19 At 7.3" Ln 25 Col 9 70%

Contents

Overview 1

Introduction........1
Why People Resist Owning a Dog or Cat........1
Uses For the ElectroPet........2
Leisure Activities........2
Working Pets........2

Your Pet's Databases 3

Naming Your New Friend........3
Friends Database........4

Your Pet's Personality 7

Companionship........7
Adjusting Your Pet's Companionship Setting........7
Playfulness........8
Protection Settings........9
Default Protection Settings........9
Adjusting Your Pet's Protection Setting........9

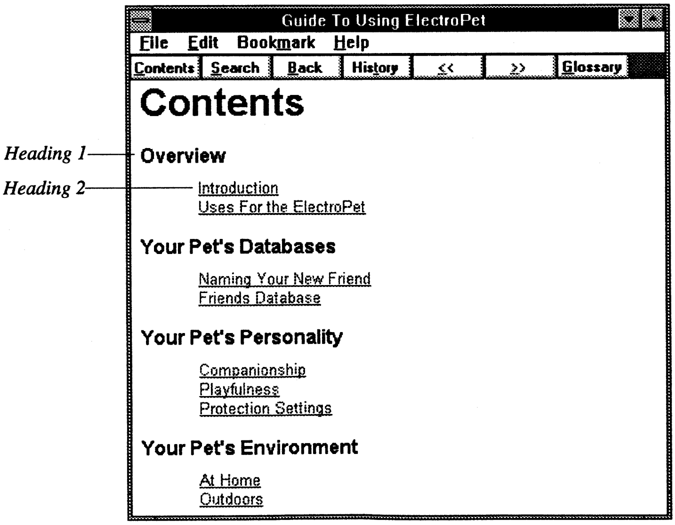
Your Pet's Environment 11

At Home........11
Sleeping........11
Outdoors........12
Walking The ElectroDog........12
Backyard Fun........12

Glossary of Terms 13

Index 15

... and if the Conversion Rules are set to show Heading 1 and 2 on the Contents Screen, Doc-To-Help will create a main help screen that looks like this:


Guide To Using ElectroPet
File Edit Bookmark Help
Contents Search Back History << >> Glossary
Heading 1
Heading 2

Contents

Overview

Introduction
Uses For the ElectroPet

Your Pet's Databases

Naming Your New Friend
Friends Database

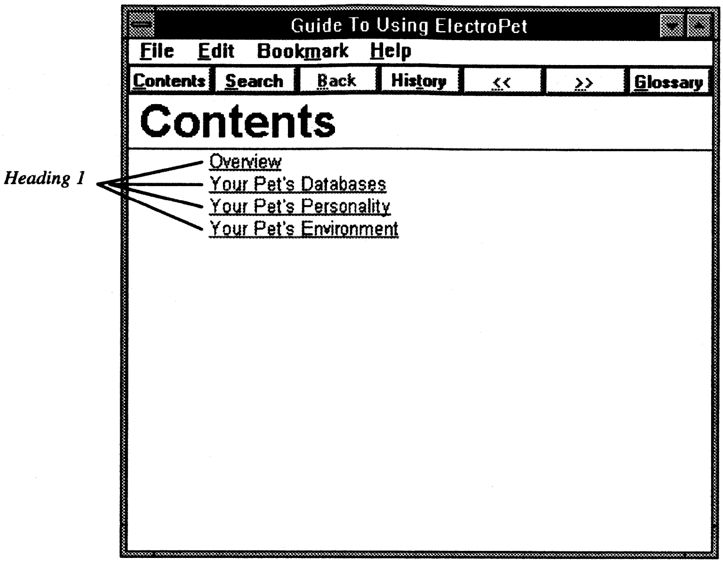
Your Pet's Personality

Companionship
Playfulness
Protection Settings

Your Pet's Environment

At Home
Outdoors

If the Conversion Rules are set to show Heading 1 only on the Contents Screen, the main help screen will look like this:

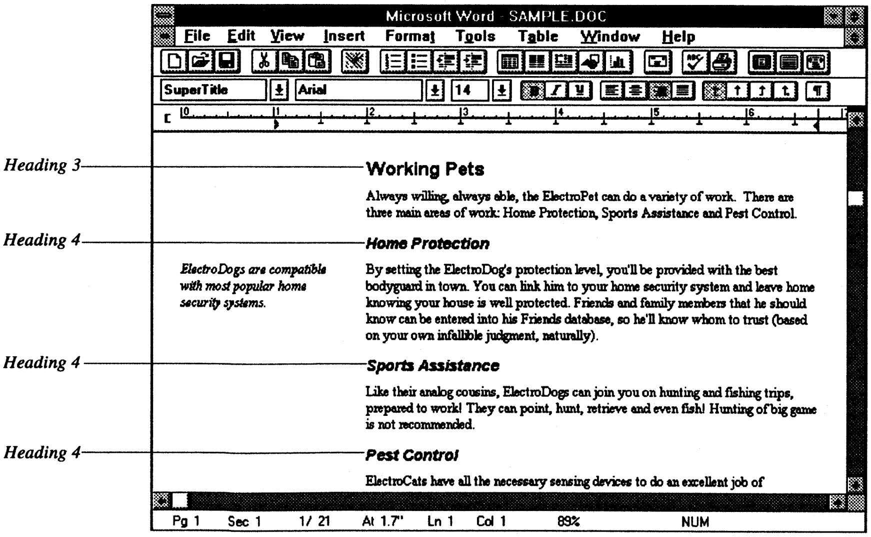


Contents

Overview
Your Pet's Databases
Your Pet's Personality
Your Pet's Environment

Topic Screens Are Based on Heading Styles

The Help file is simply a collection of topic screens. When you run Help, you move through the Help file by jumping from topic to topic. Doc-To-Help builds topic screens based on the Heading styles in your document. Each paragraph formatted in a Heading style and the text that follows it will be converted to a topic screen. Heading styles that are a level lower will appear in the Related Topics list at the bottom of the topic screen. In the example below, the page from the sample manual will be converted to *four* separate topic screens:



Working Pets

Always willing, always able, the ElectroPet can do a variety of work. There are three main areas of work: Home Protection, Sports Assistance and Pest Control.

Home Protection

ElectroDogs are compatible with most popular home security systems.

By setting the ElectroDog's protection level, you'll be provided with the best bodyguard in town. You can link him to your home security system and leave home knowing your house is well protected. Friends and family members that he should know can be entered into his Friends database, so he'll know whom to trust (based on your own infallible judgment, naturally).

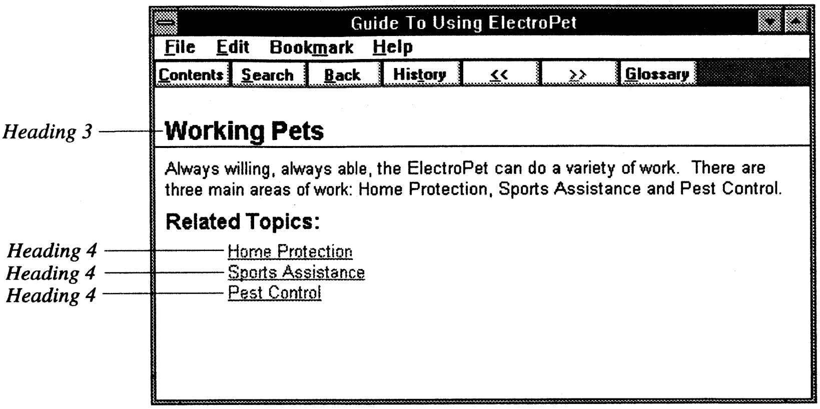
Sports Assistance

Like their analog cousins, ElectroDogs can join you on hunting and fishing trips, prepared to work! They can point, hunt, retrieve and even fish! Hunting of big game is not recommended.

Pest Control

ElectroCats have all the necessary sensing devices to do an excellent job of

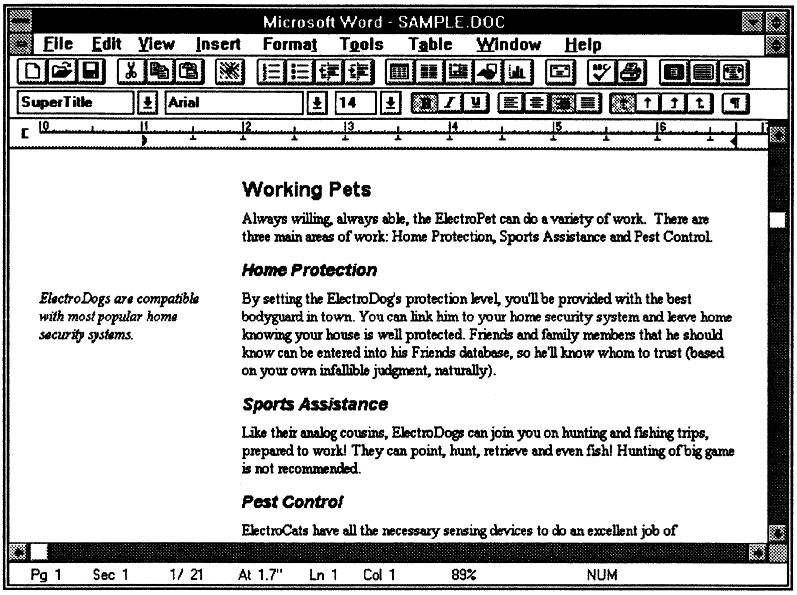
The first screen will be "Working Pets" which was originally formatted in the Heading 3 style. The paragraphs that follow in the next level down (Heading 4) are "Home Protection," "Sports Assistance" and "Pest Control." They will each be converted into separate topic screens available as jumps from the Related Topics list on the "Working Pets" topic screen:



Always willing, always able, the ElectroPet can do a variety of work. There are three main areas of work: Home Protection, Sports Assistance and Pest Control.

Margin Notes and Glossary Terms Become Definitions

Margin notes (known as captions) are converted to popup definitions in the Help file. Captions are linked to text in the document. When you click on the text the caption is linked to in the help screen, it displays the caption. Glossary Terms are also converted to popup definitions. When you click on the glossary term in the help screen, the corresponding definition from the Glossary of Terms will appear in the popup definition box. In the example below, he margin note that begins "ElectroDogs are compatible..." is linked to the text "home security system" in the paragraph next to it.


Microsoft Word - SAMPLE.DOC
File Edit View Insert Format Tools Table Window Help
SuperTitle
Arial
14
Pg 1 Sec 1 1/ 21 At 1.7" Ln 1 Col 1 89% NUM

Working Pets

Always willing, always able, the ElectroPet can do a variety of work. There are three main areas of work: Home Protection, Sports Assistance and Pest Control.

Home Protection

ElectroDogs are compatible with most popular home security systems.

By setting the ElectroDog's protection level, you'll be provided with the best bodyguard in town. You can link him to your home security system and leave home knowing your house is well protected. Friends and family members that he should know can be entered into his Friends database, so he'll know whom to trust (based on your own infallible judgment, naturally).

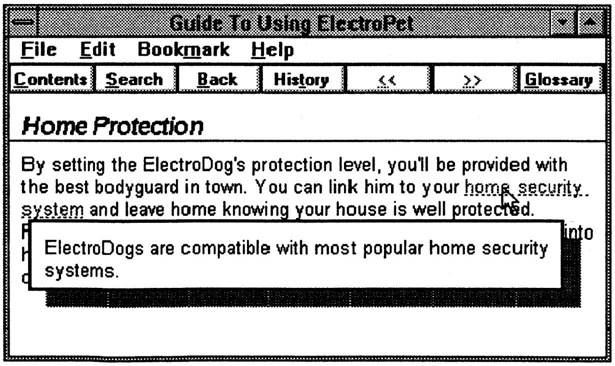
Sports Assistance

Like their analog cousins, ElectroDogs can join you on hunting and fishing trips, prepared to work! They can point, hunt, retrieve and even fish! Hunting of big game is not recommended.

Pest Control

ElectroCats have all the necessary sensing devices to do an excellent job of

Doc-To-Help will convert that margin note to a popup definition in the "Home Protection" topic screen. Clicking on the words "home security system" in this screen will display the text of the margin note.


Guide To Using ElectroPet
File Edit Bookmark Help
Contents Search Back History << >> Glossary

Home Protection

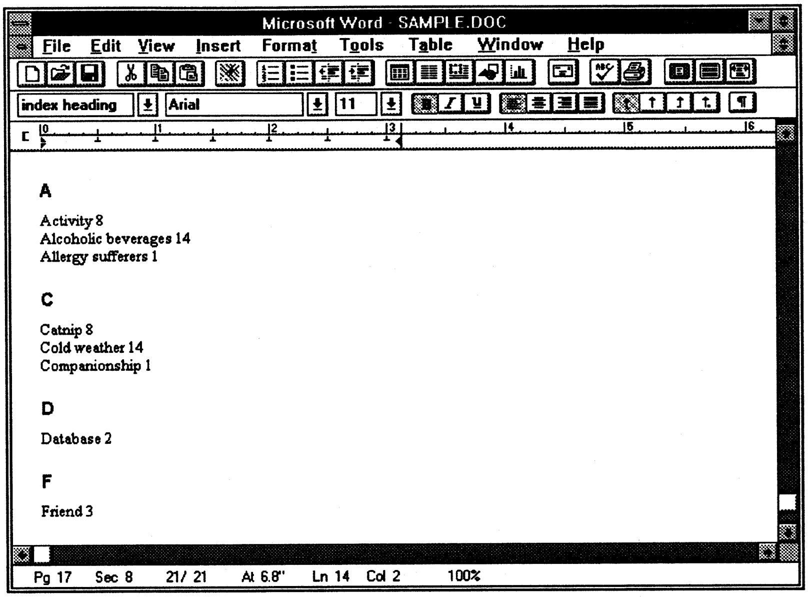
By setting the ElectroDog's protection level, you'll be provided with the best bodyguard in town. You can link him to your home security system and leave home knowing your house is well protected.

ElectroDogs are compatible with most popular home security systems.

Index Tags and Heading Text Become Search Keywords

Just as the index provides an easy way to find something in the manual, the list of keywords displayed when you click Search in Help provide an easy way to jump to specific topics. Doc-To-Help creates this search list based on the manual's index tags and heading text.

In the index below you'll notice that "Allergy Sufferers" points to page one. On page one, you'll find a reference to allergy sufferers under the heading "Why People Resist Owning a Dog or Cat."


Microsoft Word - SAMPLE.DOC
File Edit View Insert Format Tools Table Window Help
index heading
Arial
11
Pg 17 Sec 8 21/ 21 At 6.8" Ln 14 Col 2 100%

A

Activity 8
Alcoholic beverages 14
Allergy sufferers 1

C

Catnip 8
Cold weather 14
Companionship 1

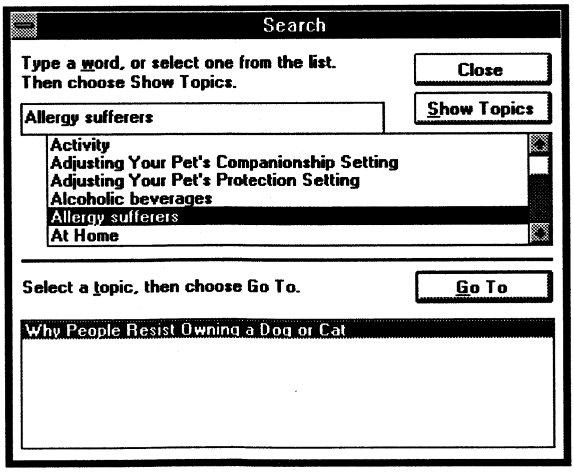
D

Database 2

F

Friend 3

Doc-To-Help will create a search list that contains the same index tags in the manual's index plus the text of all headings found in the manual:



Type a word, or select one from the list.
Then choose Show Topics.

Allergy sufferers

Activity
Adjusting Your Pet's Companionship Setting
Adjusting Your Pet's Protection Setting
Alcoholic beverages
Allergy sufferers
At Home

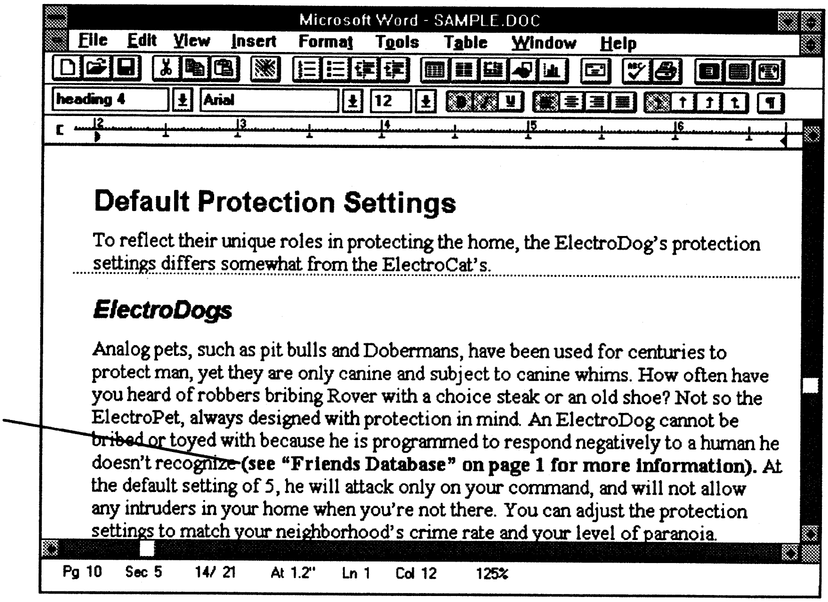
Select a topic, then choose Go To.

Why People Resist Owning a Dog or Cat

In this example, we've chosen "Allergy Sufferers" and clicked **Show Topics**. The topics list contains "Why People Resist Owning a Dog or Cat" which is the topic screen that contains a reference to allergy sufferers. Clicking **Go To** will jump you to that screen.

Cross References Are Converted to Jumps

Doc-To-Help automatically converts cross references and page references into jump text. The page below contains a cross reference to the "Friends Database":



Microsoft Word - SAMPLE.DOC
File Edit View Insert Format Tools Table Window Help
heading 4
Arial
12
Pg 10 Sec 5 14/ 21 At 1.2" Ln 1 Col 12 125%

Default Protection Settings

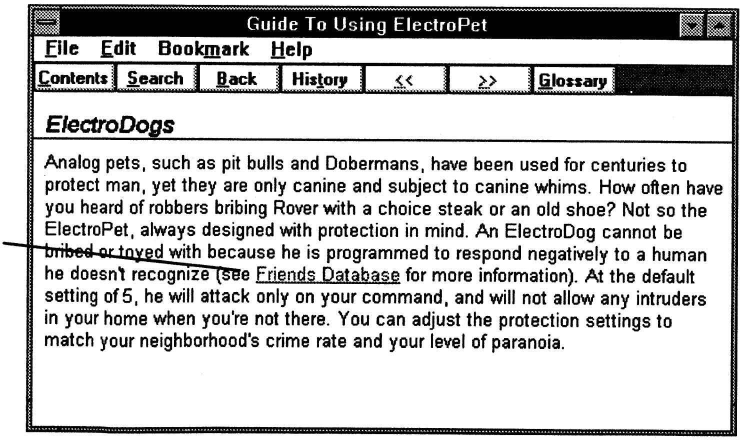
To reflect their unique roles in protecting the home, the ElectroDog's protection settings differs somewhat from the ElectroCat's.

ElectroDogs

Analog pets, such as pit bulls and Dobermans, have been used for centuries to protect man, yet they are only canine and subject to canine whims. How often have you heard of robbers bribing Rover with a choice steak or an old shoe? Not so the ElectroPet, always designed with protection in mind. An ElectroDog cannot be bribed or toyed with because he is programmed to respond negatively to a human he doesn't recognize **(see "Friends Database" on page 1 for more information).** At the default setting of 5, he will attack only on your command, and will not allow any intruders in your home when you're not there. You can adjust the protection settings to match your neighborhood's crime rate and your level of paranoia.

Cross Reference

The heading "ElectroDogs" and the text that follows it will be converted to a help screen that looks like this (notice that the cross reference has been converted to the jump "Friends Database"):



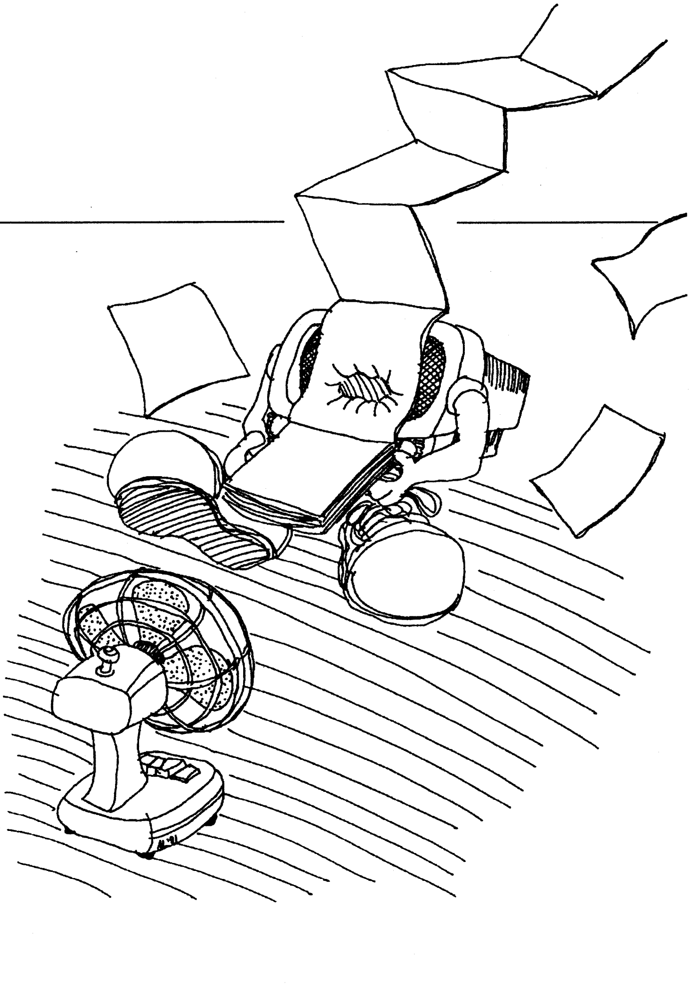
Guide To Using ElectroPet
File Edit Bookmark Help
Contents Search Back History << >> Glossary

ElectroDogs

Analog pets, such as pit bulls and Dobermans, have been used for centuries to protect man, yet they are only canine and subject to canine whims. How often have you heard of robbers bribing Rover with a choice steak or an old shoe? Not so the ElectroPet, always designed with protection in mind. An ElectroDog cannot be bribed or toyed with because he is programmed to respond negatively to a human he doesn't recognize (see Friends Database for more information). At the default setting of 5, he will attack only on your command, and will not allow any intruders in your home when you're not there. You can adjust the protection settings to match your neighborhood's crime rate and your level of paranoia.

Hypertext Jump

Working with Long Documents



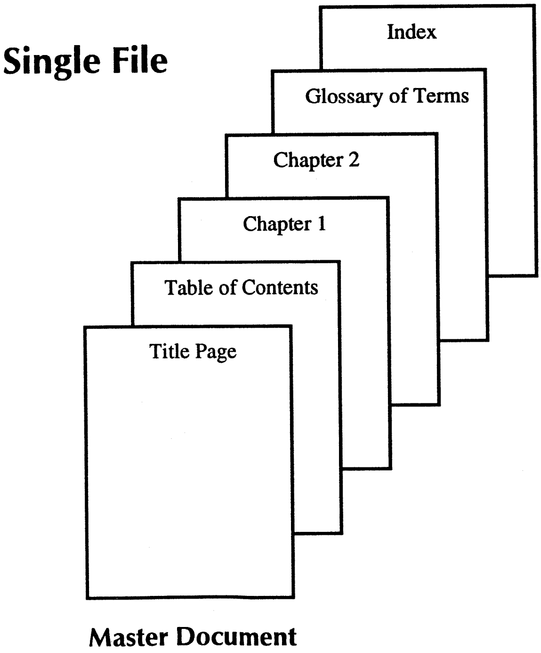
Working With Long Documents

Overview

If the manual you will be creating is long (over 200 pages) you'll probably want to divide the manual into small, manageable files as opposed to working with one unwieldy file. In this section we'll discuss how to create a multifile project from scratch, how to divide an existing manual into smaller chunks and how a group of people can use Doc-To-Help to create a single Doc-To-Help Project.

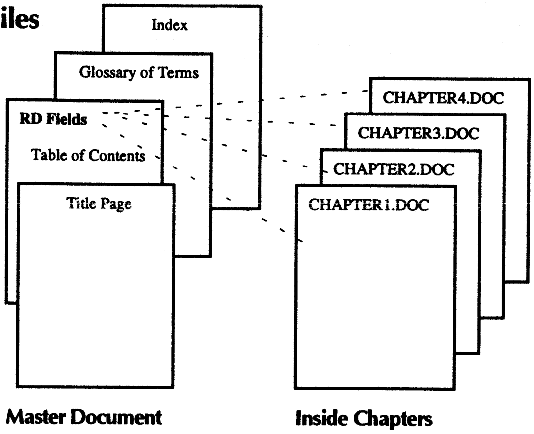
How Doc-To-Help Treats Multiple-File Projects

When working with a single file, Doc-To-Help places everything that comprises the manual — the title page, the table of contents, the chapters, the glossary of terms and the index — into one master document, as shown below.



When working with multiple files, Doc-To-Help places the title page, table of contents, glossary of terms and index in the master document, but puts the "meat" of your manual into separate chapters (as shown below).

Multiple Files



RD stands for "Referenced Document."

Doc-To-Help updates the master document's index and table of contents by using Word's RD fields, which allow you to reference heading information, page numbering information and index entries contained in external documents. You do not need to know how to use these RD fields; Doc-To-Help will take care of that for you.

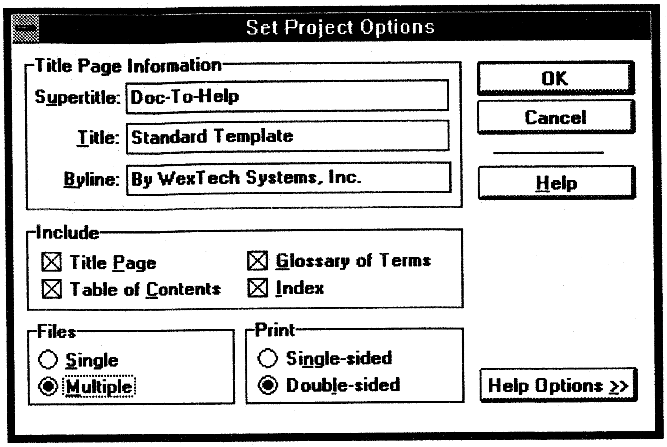
Doc-To-Help also lets you control how many documents can be open at a given time. This is especially important when you are indexing, already a memory-intensive procedure, because too many open files can drain memory. Use the **Tools Doc-To-Help Options** command to set the maximum number of documents that may be open at any one time. (See "Tools Doc-To-Help Options" on page 237 for more information.)

Creating a Multifile Project from Scratch

The process of creating a multifile project is essentially the same as with a single file manual, except that Doc-To-Help will ask you a few more questions and require you to save your work immediately.

To Create a Multifile Project

1. From the **File** menu, choose **New**.
2. When the File New dialog box appears, choose one of the Doc-To-Help manual templates (**D2H_NORM**, **D2H_SIDE** or **D2H_SMAL**) and click **OK**.
3. The Set Project Options dialog box will appear, as shown on the next page.



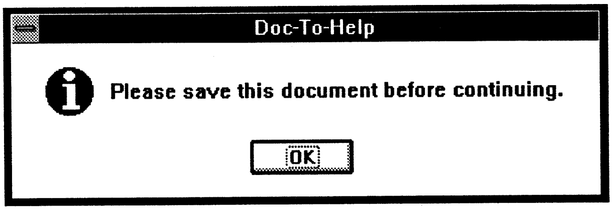
Title Page Information
Supertitle: Doc-To-Help
Title: Standard Template
Byline: By WexTech Systems, Inc.

Include
☒ Title Page ☒ Glossary of Terms
☒ Table of Contents ☒ Index

Files
○ Single
● Multiple

Print
○ Single-sided
● Double-sided

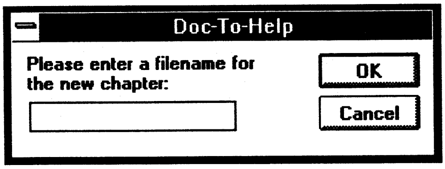
4. Enter a **Supertitle**, **Title** and **Byline** and make sure **Multiple** is selected before clicking **OK**. The following message box will appear:



Please save this document before continuing.

A Doc-To-Help project contains a DOC2HELP.INI file, the master document, any inside documents and files created when you make the project into Help. Each project must be kept in a separate project directory.

5. Click **OK**. The **File Save** dialog box will appear.
6. Enter a name and click **OK**. At this point you will have just saved the master document (which only contains a title page, table of contents, glossary and index). The New Chapter dialog box will then appear, as shown below.



Please enter a filename for the new chapter:

7. Enter a filename for the chapter document and click **OK**. Doc-To-Help will display the chapter document you just created.

Adding More Chapters

As with single file manuals, you add new chapters using the Insert New Chapter command. However, unlike the single file version (where choosing this command inserts a section break), in multifile projects choosing this command will present the New Chapter dialog box and create a new file.

New chapters will be based on a different template from master documents. If your master document is based on D2H_NORM (for standard manuals), the inside chapters will be based on D2H_IN. If the master document is based on D2H_SIDE (for sidehead manuals), the inside chapters will be based on D2H_INSD. If the master document is based on D2H_SMAL (for manuals that will be printed on smaller paper), the inside chapters will be based on D2H_INSM. (For more information, see "Doc-To-Help Templates" on page 195.)

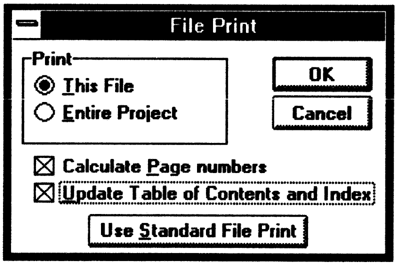
How Your Life Will Be Different

Overview

For the most part, the commands for multiple file manuals are the same as for single file manuals, with some notable exceptions: printing, inserting hypertext links, setting Doc-To-Help Options, repairing the manual, viewing markings and organizing chapters.

Printing

If you are working with a multifile project, choosing **File Print** will present the Doc-To-Help file print dialog box, as shown below.



File Print

Print
● This File
○ Entire Project

☒ Calculate Page numbers
☒ Update Table of Contents and Index

Choosing **This File** will print the file you are currently working on while choosing **Entire Project** will print the master document and all files that make up your manual.

If you've used RD fields without Doc-To-Help, you may have noticed that *you* have to keep track of the page numbering for each document. That is, if the first chapter ends on page 14, you will have to change the page numbering of the second chapter so that it starts on page 15. Choosing **Calculate Page Numbers** will do this for you by opening each file and resetting the page numbering.

Choosing **Update Table of Contents and Index** will make Doc-To-Help update the TOC and INDEX fields in the master document.

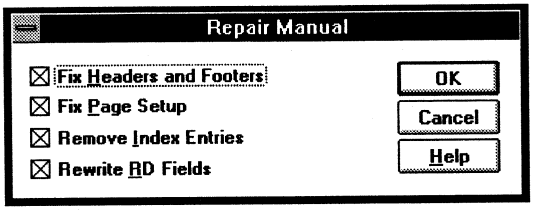
Pressing **Use Standard File Print** bypasses these options and displays Word's built-in print dialog box.

Tools Doc-To-Help Options

Word for Windows allows you to have up to 9 documents open at the same time. If you are working with large files, however, you may run out of memory even if you have fewer than 9 documents open. Because the process of indexing, making into Help, and working with a multifile project can put a drain on memory, Doc-To-Help lets you control how many files can be open while Doc-To-Help commands are running. The default is 5 files, but you can change this with **Tools Doc-To-Help Options.** (See "Tools Doc-To-Help Options" on page 237.)

Tools Repair Manual

You can correct problems with RD fields by using **Tools Repair Manual**. Choosing this command from a multifile project will display the **Repair Manual** dialog box as shown below.



Repair Manual

☒ Fix Headers and Footers
☒ Fix Page Setup
☒ Remove Index Entries
☒ Rewrite RD Fields

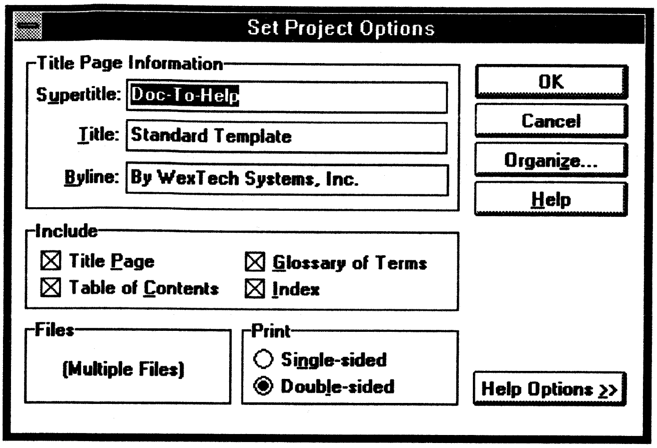
Choosing **Rewrite RD Fields** will delete existing RD fields and update them to reference the correct inside chapter files. RD fields are used to compile the Index and Table of Contents in your master document. This option is only available from the master document in a multifile project.

View Doc-To-Help Markings

If you choose **View Doc-To-Help Markings** with a multifile project, Doc-To-Help will ask you if you want to view manual/help markings for the file you're working with or for all files that comprise your manual. (See "View Doc-To-Help Markings" on page 207.)

Organizing Chapters

You determine how you want your inside chapters organized using the **Format Set Project Options** command. Choosing this command with a multifile project will present a slightly different version of the single-file Set Project Options dialog box, as shown on the next page.



Title Page Information

Supertitle: Doc-To-Help

Title: Standard Template

Byline: By WexTech Systems, Inc.

Include

☒ Title Page ☒ Glossary of Terms
☒ Table of Contents ☒ Index

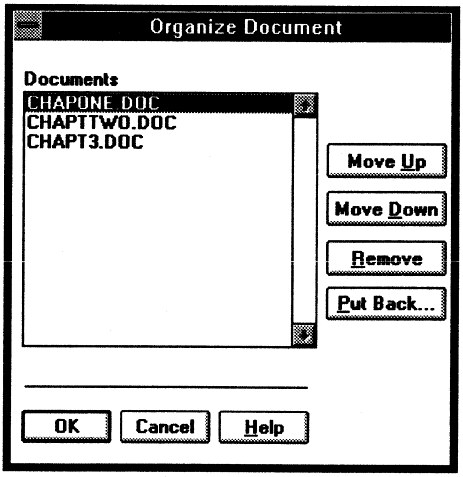
Files

(Multiple Files)

Print

○ Single-sided
● Double-sided

Pressing **Organize** will display the Organize Document dialog box, as shown below.



Documents

CHAPONE.DOC
CHAPTTWO.DOC
CHAPT3.DOC

The Organize Document dialog box displays the order in which all the inside chapters of your manual will be printed.

Click **Move Up** to move the chapter selected in the **Documents** list box up in the list.

Click **Move Down** to move the chapter selected in the **Documents** list box down in the list.

Click **Remove** to remove the chapter selected in the **Documents** list box. You will be prompted to delete the document from the disk or save a copy of it in a subdirectory called REMOVED. Doc-To-Help creates the REMOVED directory underneath the project directory.

Click **Put Back** to bring back chapters from the REMOVED subdirectory

Note: You can incorporate an existing chapter into a project by copying it to the REMOVED directory and then bringing it back with **Put Back**. Doc-To-Help will verify that the file is based on the correct inside template. If it isn't, Doc-To-Help gives you the option to attach it or cancel. (For more information see "Converting Existing Documents" on page 83.)

How to Split a Large Document into Smaller Pieces

What we've looked at so far is all fine and good if you remember to choose Multiple from the start, but what if you've already created a single-file manual that's getting unwieldy? Fortunately, you can choose Multiple even after you've already been working with a manual for a while, and Doc-To-Help will break the manual into separate files.

Warning: If your single-file manual contains hypertext links, you will need to create the links again after Doc-To-Help converts the manual into a multifile project. Also, Word for Windows will not allow to you create cross references/page references to text not contained in the file you are editing.

To Split a Long Document into Separate Files

1. Choose **Format Set Project Options** from the menu bar.
2. When the dialog box appears, choose **Multiple** and click **OK**. Doc-To-Help will determine how many chapters your manual contains and ask you to give each chapter a unique filename.

Workgroups

Having Several Authors Working on One Manual

Doc-To-Help has been designed so that a group can work together to create one manual. For this example, let's suppose you're managing a small group of people who will be helping you write a lengthy manual. The steps you will need to perform are as follows:

1. Create a new, multiple-file manual (see "Creating a Multifile Project from Scratch" on page 58).
2. Add as many chapters as are needed, using the **Insert New Chapter** command.
3. Give each of your co-authors a copy of the chapters they will be working on, along with the "inside" version of your manual's template. (That is, if your master document is based on D2H_NORM, make sure your co-authors can access D2H_IN. If your master document is based on D2H_SIDE, make sure your co-authors can access D2H_INSD. If your master document is based on D2H_SMAL, make sure they can access D2H_INSM.) Your co-authors will also need to have access to D2HELPER.EXE and DOC2HELP.DLL. Do *not* give your co-authors a copy of the DOC2HELP.INI file.

People working on inside chapters only won't be able to change the fonts globally, add a term to the glossary, build an index, and so on.

4. The inside template should be copied to the template directory on the co-authors' machines.
5. When your co-authors have finished their work, copy their files into the project directory where the master document is stored.

Breaking a Chapter Across More Than One Inside File

Occasionally you may need to split a chapter across more than one inside file. This may happen if the chapter is very large or if more than one person would like to work on that chapter simultaneously. You break a chapter into several files by following these guidelines:

- The file containing the actual chapter heading should precede the other inside files for that chapter in the Organize Chapters list box.
- The first paragraph in each inside file must be formatted in Heading 1, Heading 2, Heading 3 or Heading 4 style.
- If the inside file does not contain a Heading 1 style on the first page, the footer will display "Error! Style not defined" instead of the Heading 1 text, so you must edit the footer and replace the Styleref field with the text you want to appear in the footer.
- If you want chapter numbers displayed in the table of contents, index or footers, make sure that there is a sequence field before the first paragraph in each of the inside files. The "\r" switch in each file should reference the same chapter number. For example, if you've split Chapter Five into three files, the sequence field preceding the first paragraph in each of the three files should look like this:

  ```
  {seq Chapter \h \r5}
  ```

  (See "Displaying Chapter Numbers with Page Numbers" on page 75 for more information.)

Indexing

Indexing

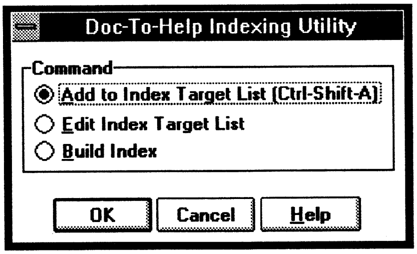
Overview

The Doc-To-Help indexing facility will appeal to both the indexing tyro and tyrant. The novice will be able to create a simple index with little or no practice and the expert will be able to build comprehensive indexes much faster (and with much less effort) than when using Word's standard indexing commands.

Creating a good index for your manual is particularly important as your document's index will seed the online help keyword search facility.

Steps Involved in Creating an Index

The indexing commands are all accessed through the **Tools Indexing** dialog box (shown below). The steps involved in creating an index are as follows:


Doc-To-Help Indexing Utility
Command
● Add to Index Target List (Ctrl-Shift-A)
○ Edit Index Target List
○ Build Index
OK
Cancel
Help

1. Using the **Add to Index Target List** command (**CTRL+SHIFT+A**), create a list of all words and phrases you want Doc-To-Help to look for when it builds the index. The words or phrases to search for are called **Index Targets**.

Note: Do not use square brackets "[ ]" in index target text.

2. Using the **Edit Index Target List** command, tell Doc-To-Help what you want it to do when it finds one of these Index Targets. Doc-To-Help can insert one or more **Index Tags** at that point in the document. For example, you can search for the Index Target "Football," and assign the Index Tags "Professional Sports," "Leisure Activities," and "Magnificent Obsession" at each instance. The resulting Index will show an entry for each of

these three tags listing the page numbers containing the word "Football."

3. Using the **Build Index** command, create the index. Based on how you edited the Index Target list in step two, Doc-To-Help will insert Index Tags, prompt you for inserting custom Index Tags, and so on.

Adding to the Index Target List

To Add to the Index Target List Using Keystrokes

Note: The limit for index targets is 115 characters.

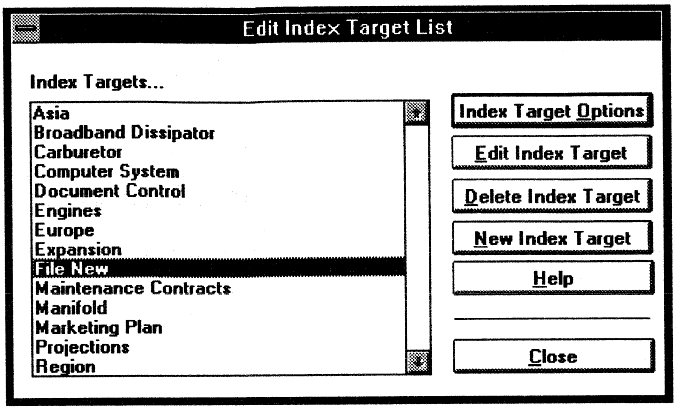
1. Highlight the word or phrase you want to add.
2. From the **Tools** menu, choose **Indexing.** Select **Add to Index Target List**, or press CTRL+SHIFT+A.
3. If the "**Prompt on Add to Target List**" option has been selected in **Doc-To-Help Options,** the Index Target Options dialog box will appear with the target you entered as a default Index Tag. (See "Editing the Index Target List" below.)

To Add to the Index Target List Using the Edit Index Target List Command

1. From the **Tools** menu, choose **Indexing.** Select **Edit Index Target List**.
2. When the Edit Index Target List dialog box appears, click **New Index Target**.
3. When the Add IndexTarget dialog box appears, type the target you want to add and click **OK**. The Index Target Options dialog box will appear with the target you entered as a default index tag. (See "Editing the Index Target List" below.)

Editing the Index Target List

Assuming you have added index targets to the index target list, choosing the **Edit Index Target List** command from the **Tools Indexing...** dialog box will produce a dialog box similar to the one shown below:



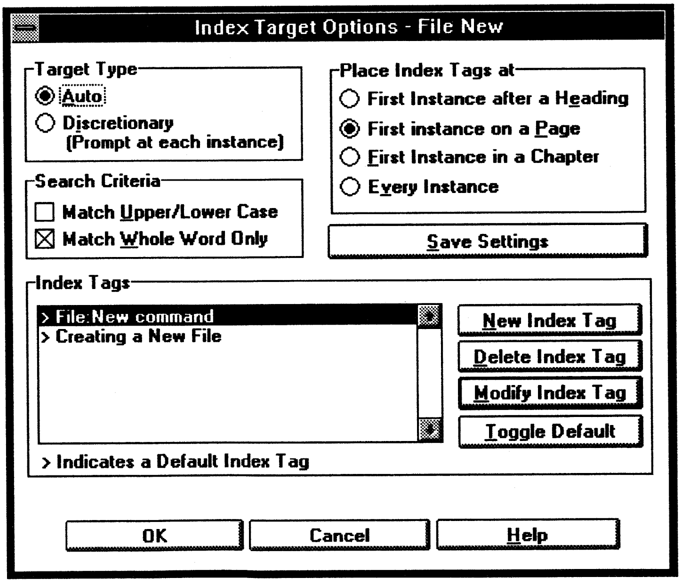
Index Targets...

Asia
Broadband Dissipator
Carburetor
Computer System
Document Control
Engines
Europe
Expansion
File New
Maintenance Contracts
Manifold
Marketing Plan
Projections
Region

The list box shows all the index targets contained in the index target list. These are the words or phrases Doc-To-Help will search for when building the index.

What Will Doc-To-Help Do When It Finds an Index Target?

You determine what Doc-To-Help will do when it finds a particular index target by selecting that target and clicking on the **Index Target Options** button. Clicking on this button will produce a dialog box similar to the one shown below.



Target Type
● Auto
○ Discretionary (Prompt at each instance)

Search Criteria
□ Match Upper/Lower Case
☒ Match Whole Word Only

Place Index Tags at
○ First Instance after a Heading
● First instance on a Page
○ First Instance in a Chapter
○ Every Instance

Index Tags
> File:New command
> Creating a New File

> Indicates a Default Index Tag

Auto indexing is the default Target Type.

When you use the **Tools Build Index** command, Doc-To-Help will find the first occurrence of the words "File New" on every page and insert the index tags "File:New command" and "Creating a New File." Since auto indexing is selected, Doc-To-Help will not prompt you when it finds the words "File New" in your document.

If you want to control which index tags are placed at a particular occurrence of an index target in your manual, you should select discretionary indexing for that index target. (See "Discretionary Indexing" on page 69 for more information.)

If identical index tags are inserted on consecutive pages, the page range will be listed in your document's index as opposed to individual page references. In other words, you'll see

Creating a New File 3-5, 19, 23-4

instead of

Creating a New File 3, 4, 5, 19, 23, 24

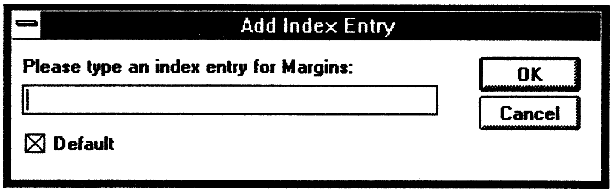
You can change this setting in the **Tools Doc-To-Help Options** dialog box. (See "Tools Doc-To-Help Options" on page 237.)

Save Settings

The **Save Settings** button sets the defaults (target type, search criteria, etc.) for any index targets you may add later.

Adding More Index Tags

You create additional index tags by pressing the **New Index Tag** button in the Index Target Options dialog box. Pressing the button will produce a dialog box similar to the one shown below.


Add Index Entry

Please type an index entry for Margins:

☒ Default

If the Index Target Type is Auto, make sure the index tags are marked as Default for the Index Target. Otherwise, they'll be ignored.

Unless you specify otherwise, each new index tag will be designated as a default index tag. The way Doc-To-Help handles default tags depends on whether the target type is Auto or Discretionary. If the Index Target Type is Auto, default tags will be automatically placed at the index target location. If the Index Target Type is Discretionary, you will be prompted to place the default tags and/or any non-default tags at the index target location (see below). So, if the Target Type is Discretionary, tags that you want placed at the target location most or all of the time should be designated as default.

Creating a Custom Index

Discretionary Indexing

Consider the dialog box shown below.

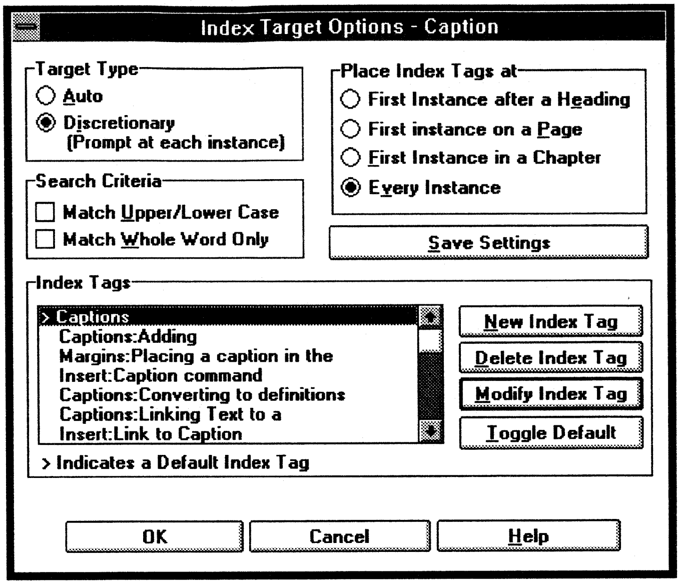
Entering a colon between words will create a multiple-level index. With discretionary indexing, the example to the right will produce an index that looks like this:

Captions 2, 9, 12, 34
  Adding 34
  Converting to definitions 9
  Linking Text to a 12

Insert
  Caption command 34
  Link to Caption 12

Margin Notes 2, 9, 34

Margins
  Placing a caption in the 34



Index Target Options - Caption

Target Type
○ Auto
● Discretionary (Prompt at each instance)

Search Criteria
□ Match Upper/Lower Case
□ Match Whole Word Only

Place Index Tags at
○ First Instance after a Heading
○ First instance on a Page
○ First Instance in a Chapter
● Every Instance

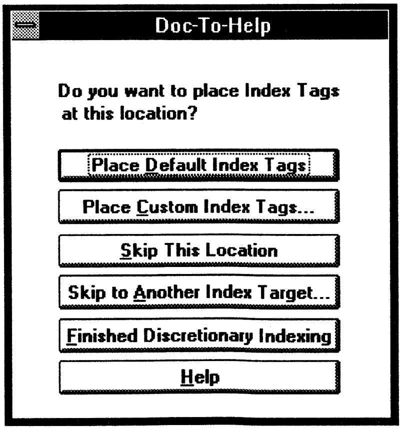
Index Tags
> Captions
Captions:Adding
Margins:Placing a caption in the
Insert:Caption command
Captions:Converting to definitions
Captions:Linking Text to a
Insert:Link to Caption

> Indicates a Default Index Tag

Since Index Target Type is Discretionary, Doc-To-Help will prompt you whenever it finds an occurrence of the word "caption" in your manual as it builds the index.

What Happens When You Build the Index

Again, since Discretionary indexing is chosen, the following dialog box will appear when Doc-To-Help finds the word "caption" after you choose **Build Index** from the **Tools Indexing** dialog box:



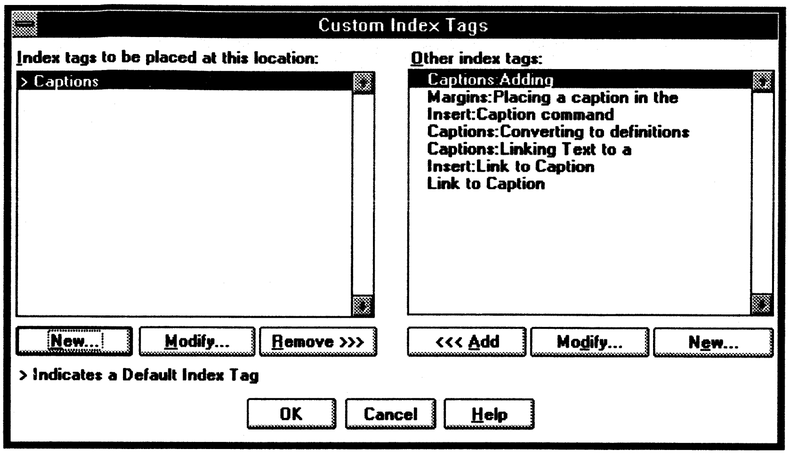
Doc-To-Help

Do you want to place Index Tags at this location?

Place Default Index Tags
Place Custom Index Tags...
Skip This Location
Skip to Another Index Target...
Finished Discretionary Indexing
Help

Actually, Doc-To-Help inserts a temporary index code which is replaced with a Word for Windows XE field when you finalize the index.

If you were to choose Place Default Index Tags, Doc-To-Help will insert the index tag "Captions" but will not insert the other tags. Choosing Place Custom Index Tags will display the following dialog box.



Index tags to be placed at this location:

> Captions

Other index tags:

Captions:Adding
Margins:Placing a caption in the
Insert:Caption command
Captions:Converting to definitions
Captions:Linking Text to a
Insert:Link to Caption
Link to Caption

> Indicates a Default Index Tag

At this point clicking on **OK** would produce the same results as if you had chosen Place Default Index Tags above; that is, only the index tag "Captions" would be inserted.

Add Button

Use the **Add** button to take one of the "sometimes used" entries in the list box on the right and move it to the list box on the left. In this example, you could add "Margins:Placing a caption in the."

Remove Button

Use the **Remove** button to prevent a particular index tag from being inserted at the current location. Pressing Remove will take the selected entry from the left list box and move it to the right list box. In this example, you would remove "Captions."

Modify Button

Use the **Modify** button to change one of the index tags. Modifying an index tag will change the way that entry is inserted throughout the document, not just at the current location. In this example, you decide during indexing that the index tag "Caption:Adding" isn't clear enough. You can modify the index tag so that it reads "Caption:Adding in Margin." Any index tags referring to "Caption:Adding" will now refer to "Caption:Adding in Margin," even ones you've already placed.

New Button

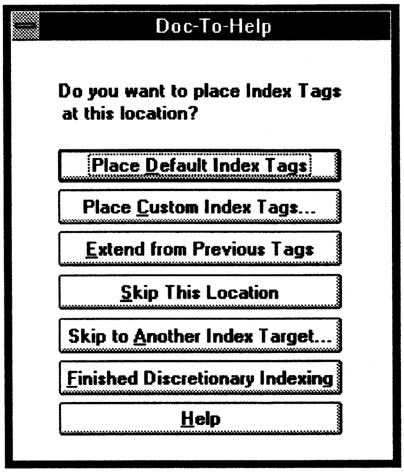
Use the **New** button to create additional index tags. You may wish to choose **Go Back To the Top** when finishing discretionary index to add this new tag at previous instances of the index target. (See "Finishing Up" on page 72.)

OK Button

Pressing the **OK** button will insert special indexing codes into your document. These codes will be replaced with standard Word for Windows index entry fields when you finalize your index.

What Happens Next

After you've placed the tag at the first index target location, Doc-To-Help will search for the next target location. If there is no other occurrence of the index target Doc-To-Help will begin searching for the next index target. If there is another occurrence of the index target, the following dialog box will be displayed:



Doc-To-Help
Place Default Index Tags
Place Custom Index Tags...
Extend from Previous Tags
Skip This Location
Skip to Another Index Target...
Finished Discretionary Indexing
Help

Do you want to place Index Tags at this location?

You can choose to place default or custom index tags at this location.

Extend From Previous Tags

This will extend the index entry from the previous location to the current location. For example, let's say pages 5 through 9 in your manual contain information about "Captions." You can place the first index tag on page 5. Then when Doc-To-Help finds the occurrence of the word "caption" on page 9 choose **Extend From Previous Tags**. The entry in the index will look like this:

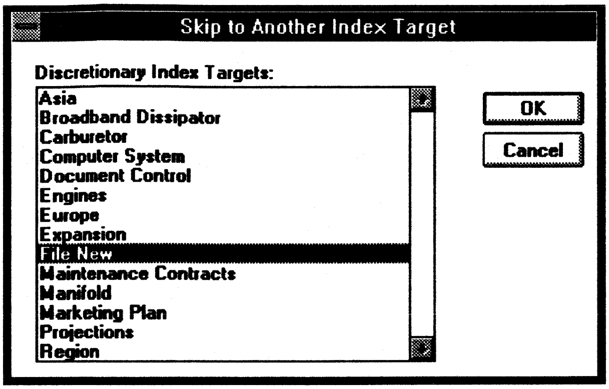
Captions 5-9

Skip This Location

Choosing this command will cause Doc-To-Help to search for the next occurrence of the index target.

Skip to Another Index Target

If you previously suspended indexing, you can begin again from where you left off. Choosing this command will display the following dialog box:


Skip to Another Index Target

Discretionary Index Targets:

Asia
Broadband Dissipator
Carburetor
Computer System
Document Control
Engines
Europe
Expansion
File New
Maintenance Contracts
Manifold
Marketing Plan
Projections
Region

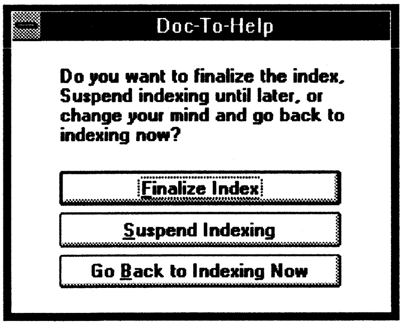
Choose the Index Target you'd like to begin with and click **OK**.

Finished Discretionary Indexing

This will end discretionary indexing and give you the option to finalize the index, suspend indexing or go back to indexing now.

Finishing Up

When Doc-To-Help has found the last occurrence of the last index target that required prompting, or if you've chosen **Finished Discretionary Indexing**, the following dialog box will appear:


Doc-To-Help

Do you want to finalize the index, Suspend indexing until later, or change your mind and go back to indexing now?

Finalize Index

Pressing **Finalize Index** will replace temporary index codes with Word index field codes and process any Auto index targets. You can see the fruits of your efforts by clicking in the index at the end of your document and pressing the Update Fields key (**F9**). If this is the first time you are building the index, click in the line that says **Error! No index tags found**.

Suspend Indexing

Pressing **Suspend Indexing** leaves everything intact (including the temporary index codes) and gives you the chance to build the index at a later session by choosing **Build Index** again.

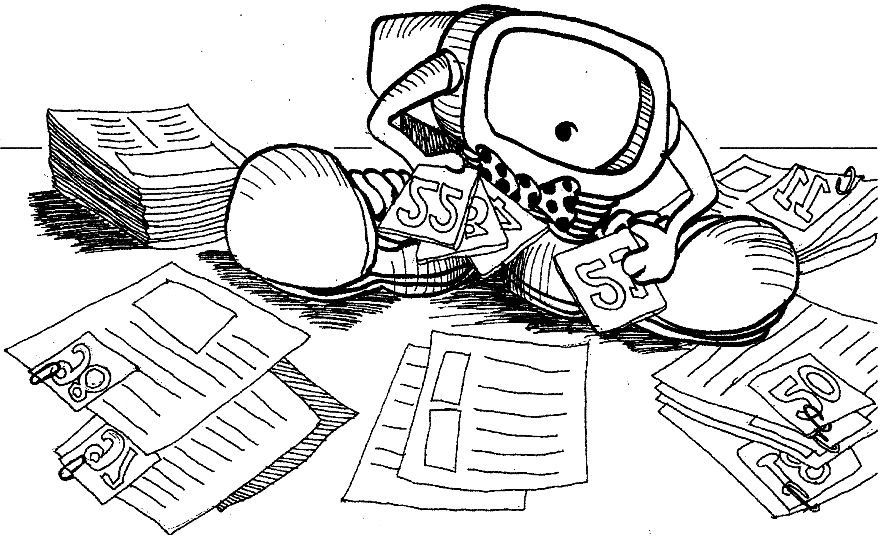
Go Back to Indexing Now

Clicking **Go Back to Indexing Now** will take you back to where you left off building the index. You may wish to do this if you've created new tags during **Build Index**.

Can the Index Be Rebuilt?

Yes. If you build the index, but later make changes to the index list, you can build the index again by choosing **Tools Build Index**. Doc-To-Help will insert new index tags. If you want to remove obsolete index tags, use the Tools Repair Manual command before building the index.

Displaying Chapter Numbers with Page Numbers



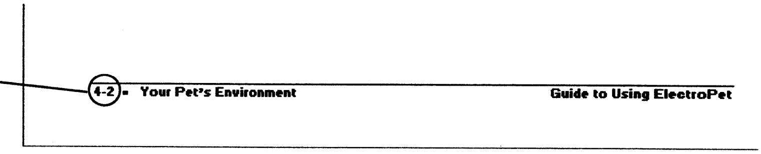
Displaying Chapter Numbers with Page Numbers

Overview

Based on your design needs,you may want to include chapter numbers as well as page numbers in your footers. Instead of continuing page numbering from the previous chapter, you can begin numbering the first page of each chapter as page one. References in the Table of Contents and the Index will be chapter number, a hyphen and the page number.

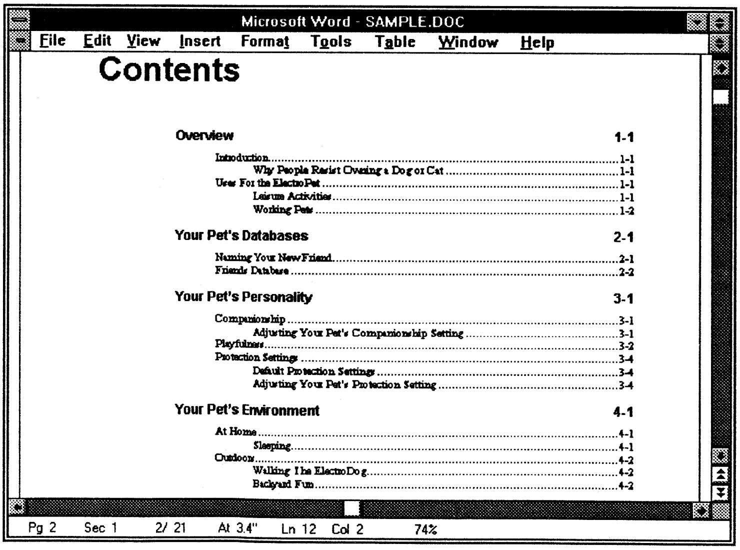
The footer might look something like this:

Footer Showing Page and Chapter Numbers



4-2 ▪ Your Pet's Environment   Guide to Using ElectroPet

The Table of Contents would look something like this:



Contents

How It Works

Chapter sequence page numbering is accomplished using Word for Windows' **Sequence Fields**. A Sequence field is a special instruction which allows Word for Windows to keep track of parts of your document, like chapters, tables or pictures, and number them automatically. There are four steps for using chapter sequence page numbering, which we'll go over in detail:

1. Adding a Sequence field to the beginning of each chapter, and to the beginning of the Glossary and Index sections if you're using them.
2. Resetting the page numbers at the beginning of each section to begin with page one.
3. Changing the footers to indicate the chapter numbers.
4. Modifying Table of Contents and Index fields to observe the sequence fields you added in step 1.

Using chapter sequence page numbering is virtually the same in single-file and multifile documents, except for a few pitfalls which multifile users need to look out for. We'll discuss these topics in "Other Considerations" on page 81.

Numbering the Chapters

The first step is to tell Word for Windows which chapter is which. To do this, you'll need to insert a Sequence field at the beginning of each chapter.

Unless you specify otherwise, sequence fields produce a result. That is, a number will appear in your document wherever you place the sequence field. Moreover, sequence fields are designed to increment automatically from one to the next, in order to automatically number parts of your document. These defaults aren't appropriate in our case, since we only want to mark the beginning of each chapter without seeing a number at the beginning, and since, in the case of multifile projects, we need to control the numbering of chapters manually. We'll include some special switches in our sequence fields to hide the result and force the sequence to reset to the number we specify.

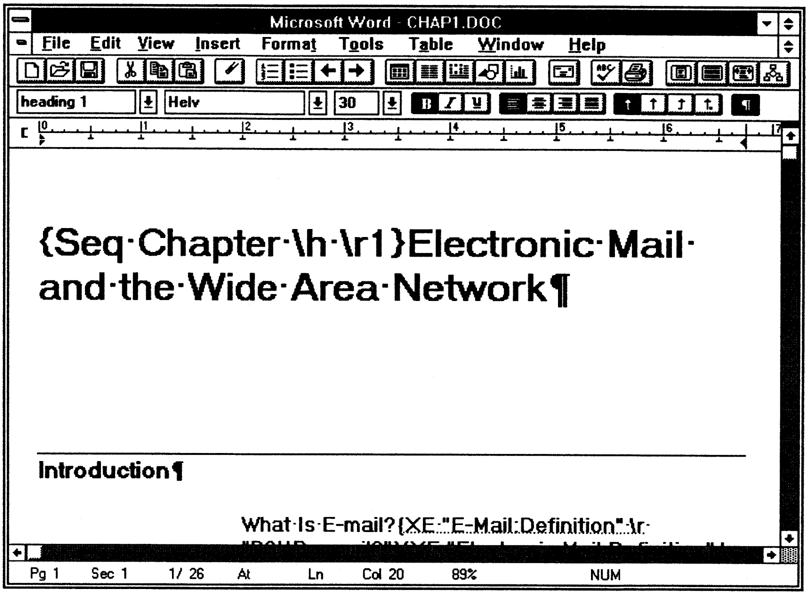
To Add the Sequence Fields

1. Go to the beginning of the first chapter. If you're numbering chapters in a multifile document, open the first *Inside* chapter, *not* the Master Document, and position the cursor at the beginning of the file.
2. Press CTRL+F9 to insert a new field. Your cursor should now be flashing between the two curly braces { } which delimit the field.

3. Type “Seq Chapter” (without the quotes) to indicate that this is a sequence field with the sequence identifier “Chapter.” The sequence identifier distinguishes this sequence from any other sequences you might be using in your document(s). In this way, Word for Windows can, for example, number chapters and tables independently.
4. Press the **spacebar** and type “\h” to hide the result of the sequence field.

You don’t need this switch in a single-file document because Word for Windows will increment the sequence fields automatically.

5. If you’re using a multifile project, press the **spacebar** again and type “\r” followed by the number of this chapter. Assuming that this is the first chapter, you should type “\r1.” Your screen should look something like the one shown below.



{Seq·Chapter·\h·\r1}Electronic·Mail·and·the·Wide·Area·Network¶

Introduction¶

What·Is·E-mail?{XE·"E-Mail:Definition"·\r·

If a chapter spans more than one inside chapter document you will need a sequence field in front of the first paragraph in each inside file. Make sure the “\r” switch references the same chapter number in each of these sequence fields.

6. Repeat steps one through five for each chapter in your manual, including the Glossary and Index sections, if you’re using them.

Resetting the Page Numbers

Next, we need to restart page numbering at page one for each chapter. Page numbering is done using the **Header/Footer** command from the **View** menu.

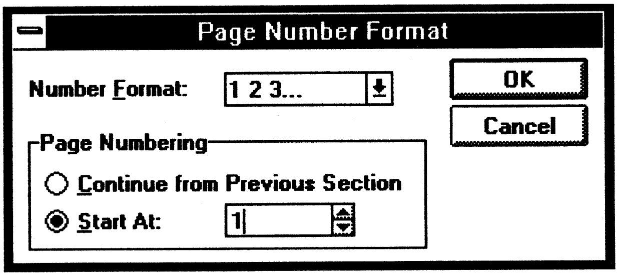
Section breaks are displayed as horizontal double lines when you're in Normal View. We suggest switching to Normal view before you reset the page numbers because the section breaks are clearly visible.

If you're using a single-file document it's important to note that page numbering is specific to each Word for Windows **section**. Sections are delimited by a **section break**, which appears on your screen as a double dotted line. If you've added chapters to your manual using some method other than Doc-To-Help's **New Chapter** command from the **Insert** menu, you may not have a section break at the beginning of each chapter. If this is the case, you should use the **New Chapter** command as described in "Insert New Chapter" on page 213 to insert the appropriate section break at the beginning of each chapter.

If you're using a multifile document, it's assumed that you've assigned one chapter per inside document. There are occasions where you may split a chapter across more than one file or put more than one chapter in a single inside file. If you've split a chapter across more than one file you should reset the page numbering on the first file only. If you've put more than one chapter in an inside file, make sure that there is a section break before each Heading 1 (except the first one in the document).

To Reset the Page Numbers

1. Go to the beginning of the first chapter in your document. If you're using a single-file document, this will be the heading that appears below the section break at the beginning of each chapter. If you're using a multifile document, this will be the top of each inside chapter document.
2. From the **View** menu, choose **Header/Footer**.
3. Click the **Page Numbers** button. The **Page Number Format** dialog box will appear on your screen.



Page Number Format

Number Format: 1 2 3...

Page Numbering

○ Continue from Previous Section

● Start At: 1

4. Click in the edit box next to the **Start At** button, and type the number 1, then click **OK**.
5. Repeat steps 1 through 4 for each chapter in your document, including the Glossary and Index sections if your manual has them.

Changing the Footers

The next step is to change the footers in your manual to reflect the chapter number as well as the page number. This step is optional. You might decide to leave the chapter number out of the footer, and place it instead at the top of your page, in the header. In any case, we'll describe how to modify the footer of your document to print a chapter number followed by a hyphen and then the page number (e.g. **3-27**).

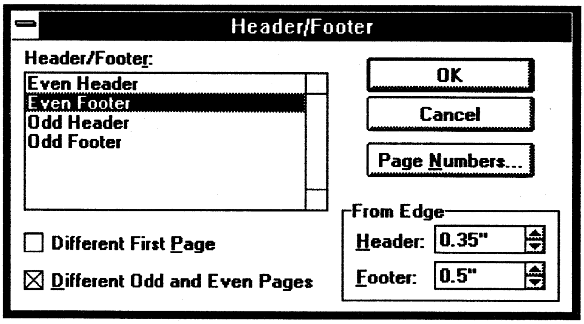
Headers and footers, like page numbers, can be different in different sections of your document. They can also be linked between sections so you don't have to manually copy your headers and footers from one section to the next every time you change them. If you're using a single-file document, all of your chapters should share the same headers and footers, so you'll only have to change the headers and footers for the Table of Contents section, the first main chapter, the Glossary section and the Index section.

To Change the Footers

1. Make sure that you're in Normal View by selecting **Normal** from the **View** menu, and move to the first chapter in your document. If you're using a multifile document, this will be the first Inside chapter document in your manual.

*You choose single- or double-sided printing using **Set Project Options** from the **Format** menu.*

2. From the **View** menu, choose **Header/Footer**. The **Header/Footer** dialog box will appear. If you've selected double-sided printing, you'll see separate entries in the list box for Even and Odd Footers.



Header/Footer

Header/Footer:

Even Header
Even Footer
Odd Header
Odd Footer

☐ Different First Page

☒ Different Odd and Even Pages

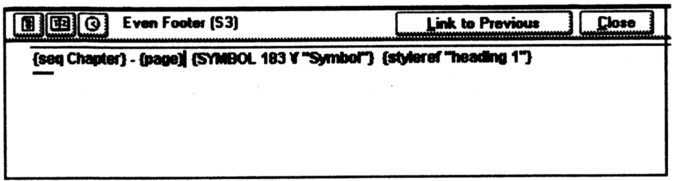
From Edge

Header: 0.35"

Footer: 0.5"

3. Select the footer, and click **OK**. If you have separate entries for Even and Odd footers, you can select either one of them.
4. Make sure Field Codes are displayed. If you don't see a check box next to the **Field Codes** command in the **View** menu, select the **Field Codes** command to turn them on.
5. Position the cursor just to the left of the **PAGE** field, and press CTRL+F9 to insert a new field.
6. Type "Seq Chapter" to indicate that this is a sequence field with the sequence identifier "Chapter."

7. Move to the right of the sequence field you've just inserted, so that the cursor is between the **Seq** and **PAGE** fields, and then type a hyphen (-).



{seq Chapter} - {page} {SYMBOL 183 \f "Symbol"} {styleref "heading 1"}

Footer with Sequence Field and Hyphen Added

8. If you have separate Even and Odd footers, repeat steps 2 through 7 for whichever footer you didn't choose the first time around.
9. Click the **Close** button to close the Footer pane.
10. If you're using a multifile document, repeat steps one through nine for each inside chapter in your manual.
11. Repeat steps one through nine for the Glossary and Index sections, if your manual has them.

Modifying the TOC and INDEX Fields

By now, your document should have properly numbered pages. The final step is to make the Table of Contents and Index fields observe the chapter number whenever you rebuild the Table of Contents and Index fields.

Word for Windows' TOC and Index fields have special switches for just this purpose. We'll show how to apply these switches to produce a table of contents and index with both chapter and page numbers.

To Include Chapter Numbers in the Table of Contents

1. Make sure Field Codes are displayed. If you don't see a check mark next to the Field Codes command in the View menu, select the Field Codes command to turn them on.
2. Position the cursor just before the curly brace at the end of the TOC Field. This is the field in your manual's Contents section that looks like this:

   {TOC \o "1-3"}

3. Press the spacebar and type "\s Chapter" so the TOC Field looks like this:

   {TOC \o "1-3" \s Chapter}

4. Turn Field Codes off. With the cursor positioned in the table of contents, press **F9** to update the Table of Contents.

To Include Chapter Numbers in the Index

1. Make sure Field Codes are displayed. If you don't see a check mark next to the Field Codes command in the View menu, select the Field Codes command to turn them on.
2. Position the cursor just after the word "index" in the Index Field. This is the field in your manual's Index section that looks like this:

   ```
   {index \h "A" \e" " \l ","}
   ```

3. Press the spacebar and type "\s Chapter" to tell Word for Windows to use the "Chapter" sequence when it builds the Index.
4. Press the spacebar again and type "\d" followed by another space, and then a colon ":" in quotes so that the Index Field looks like this:

   ```
   {index \s Chapter \d ":" \h "A" \e " " \l ","}
   ```

   The \d ":" switch at the end tells Word for Windows to use the colon as a delimiter, or separator, between the chapter and page numbers. We use a colon in the index because the hyphen is already used to indicate page ranges.
5. Turn field codes off. Press **F9** to recalculate the Index.

Other Considerations

Cross References

If you want cross references to be consistent with chapter sequence page numbering, you should change them to display the chapter and page numbers. While Doc-To-Help doesn't provide this service automatically, you can change Cross References after you've inserted them by using the following procedure.

To Use Chapter Sequence Numbering in Cross References

1. Insert the Cross Reference, if you haven't already done so, using the **Cross Reference** command from the **Insert** menu.
2. Make sure Field Codes are displayed. If you don't see a check box next to the **Field Codes** command in the **View** menu, select the **Field Codes** command to turn them on.
3. Position the cursor just before the **PageRef** field, and press **CTRL+F9** to insert a new field.
4. Type "Seq Chapter" followed by the bookmark name: this will reference the chapter number where the bookmark resides.
5. Type a hyphen between the **Seq** and **PageRef** fields. With field codes displayed, the cross reference would look something like this:

   For more information, see page {Seq Chapter BkmkName}-{PageRef BkmkName}.

   In your document it will look like this:

   For more information, see page 5-15.

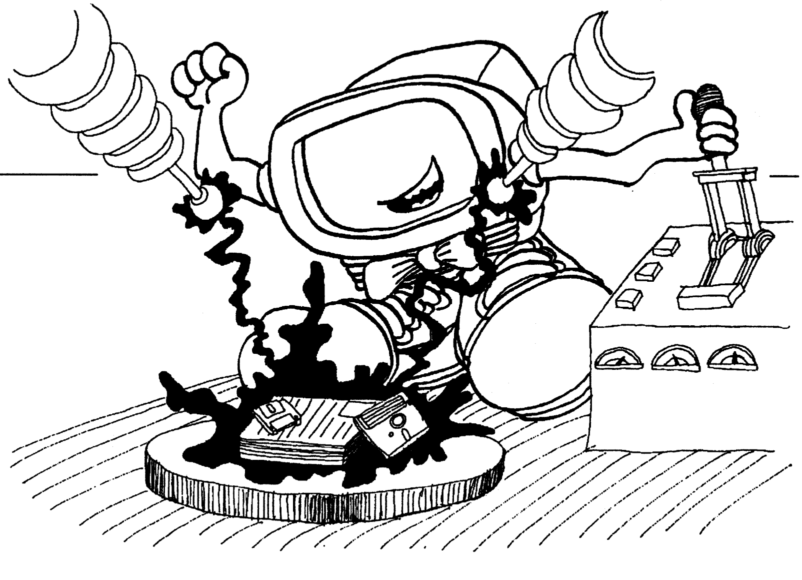
Pagination

If you're using a multifile document, you need to be careful when using the **Print** and **Update Fields** commands. With each of these commands Doc-To-Help asks you if you want to renumber the pages in your manual. If you say **Yes**, Doc-To-Help will start numbering each section where the last one left off. If you're starting each chapter at page one, you'll have to go back and reset the page numbers in **Header/Footer**.

Reorganizing the Manual

If you decide to make changes to the structure of your multifile project, you will have to manually update the **\r** switch in the sequence fields that appear at the beginning of each chapter. Otherwise, the chapter numbers in the footer, table of contents and index will be incorrect.

Converting Existing Documents



Converting Existing Documents

Overview

As long as you have Word for Windows 2.0 and Doc-To-Help you can convert almost any manual into Help, even manuals written using a word processing package other than Word. The easiest way to convert manuals into Doc-To-Help format is to create a "shell" document based on one of the Doc-To-Help templates and then insert your files into the "shell" document.

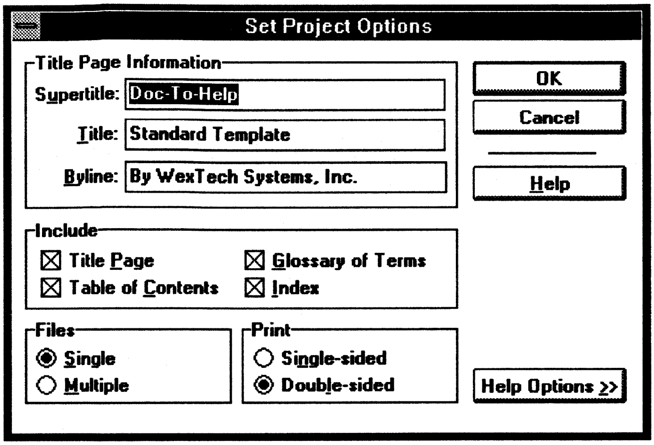
Once the files are inserted, you will need to do a certain amount of reformatting. Briefly, the steps for reformatting are as follows:

- Use Tools Repair Manual to reformat headers, footers and page layout to match the underlying Doc-To-Help template.
- Replace existing styles with Doc-To-Help styles.
- Convert tables to Doc-To-Help standard tables.
- Reformat graphics.
- Remove frames and reposition text and graphics.
- Reformat the glossary of terms.
- Remove index entries.

Before You Begin

When you create the shell document, you'll be prompted to set the project options. The settings you choose in this dialog box will determine what the shell document will look like. Even if your manual is already formatted as double-sided or contains a table of contents, index, glossary or title page, you'll still need to acknowledge those settings in the Set Project Options dialog box.

When you create a new document, the **Set Project Options** dialog box will appear as shown on the next page.



Title Page Information

Supertitle: Doc-To-Help

Title: Standard Template

Byline: By WexTech Systems, Inc.

Include

☒ Title Page ☒ Glossary of Terms

☒ Table of Contents ☒ Index

Files

● Single

○ Multiple

Print

○ Single-sided

● Double-sided

Title, Supertitle and Byline

Doc-To-Help uses this information in the footers and on the title page. When you first create the document, Doc-To-Help combines the Title and Supertitle and places it in the File Summary Info Title field. This is pulled into the footer as the document title. The Title, Supertitle and Byline will be used to create the Title Page.

Included Sections

Doc-To-Help will create sections in your shell document for Table of Contents, Index, Glossary of Terms and Title Page if you select these sections. You may already have these in your original manual but you must explicitly select them in the **Set Project Options** dialog box. After you've created the shell document, you will copy the text of your Table of Contents, Index, Glossary and Title Page into the sections Doc-To-Help creates.

Files

If you are converting a multiple file document, you should convert the master document first and then convert the inside chapters. Make sure that multiple file is selected if you are working with a multifile project.

Printing

Doc-To-Help uses this selection to determine the type of headers and footers your document will contain and what the page layout will be. Even if you've already made these choices in your original manual, you will need to make the appropriate selections in Set Project Options.

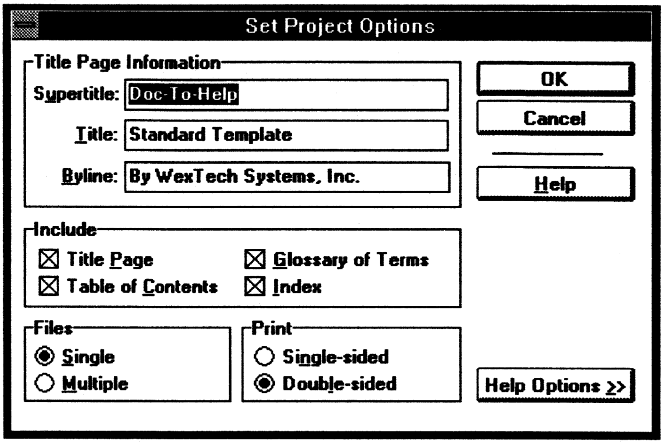
Initial Conversion

Follow the steps below to create the shell document and insert your file.

Note: If the file you want to convert is an RTF file used to create a pre-existing help file, follow the instructions for creating a new shell document then skip ahead to "Working with Pre-Existing RTF Files" on page 93.

To Convert Single File Documents

1. From the **File** menu choose **New**. When the File New dialog box appears, select the desired Doc-To-Help template. The **Set Project Options** dialog box will appear as shown below.



Set Project Options

Title Page Information

Supertitle: Doc-To-Help

Title: Standard Template

Byline: By WexTech Systems, Inc.

Include

☒ Title Page ☒ Glossary of Terms

☒ Table of Contents ☒ Index

Files

● Single

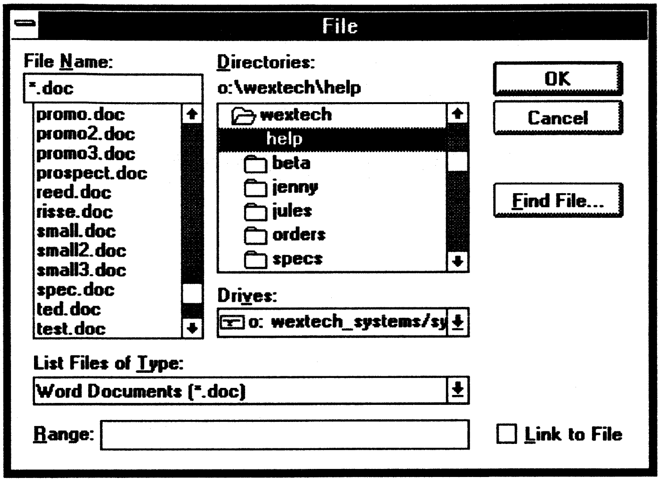
○ Multiple

Print

○ Single-sided

● Double-sided

2. Even if your document already includes a table of contents, index, glossary and title page, you should still make sure that they are selected in this dialog box. Doc-To-Help will create these sections in your document and you can copy your table of contents, index, glossary and title page to the appropriate sections. Click **OK**.
3. Highlight the "Chapter 1" place holder.
4. From the **Insert** menu, choose **File**. A dialog box similar to the following will appear.



File

File Name: *.doc

Directories: o:\wextech\help

Drives: o: wextech_systems/sy

List Files of Type: Word Documents (*.doc)

Range:

□ Link to File

5. Select the file you want to insert and click **OK**. (If the file is not a Word for Windows file, Word will present the **Convert File** dialog box. If this happens, choose the appropriate option and click **OK**.)
6. Follow the steps for reformatting your document beginning with "Using Tools Repair Manual " on page 87.

To Convert Multiple File Documents

If your document was created in separate files, you can keep it in separate files when you convert it to Doc-To-Help format.

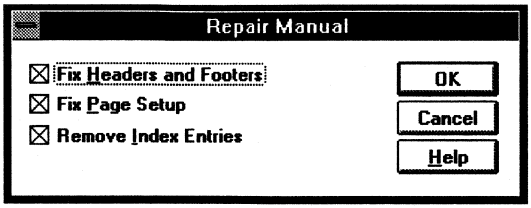
1. From the **File** menu choose **New**. Choose either D2H_NORM, D2H_SMALL or D2H_SIDE as the template.
2. In the **Set Project Options** dialog box, make sure that **Multiple** is selected in the **Files** group box. Click **OK**.
3. You will be prompted to create the first chapter file.
4. Use **Insert File** to insert the original first chapter file. Follow the steps for reformatting your document beginning with "Using Tools Repair Manual " below.
5. Use **Insert New Chapter** to create the next chapter file.
6. Repeat steps 4 and 5 until you've converted all the chapters in the project.

Note: For more information on multifile projects see "Working With Long Documents" on page 57.

Using Tools Repair Manual

After you've created the shell document and inserted your manual, you can begin the "clean up" process with Tools Repair Manual.

Tools Repair Manual adjusts the headers and footers and page setup of your document to match the underlying Doc-To-Help template. It also lets you remove existing index entries from the document. When you choose this command the **Tools Repair Manual** dialog box will appear, as shown below.



☒ Fix Headers and Footers
☒ Fix Page Setup
☒ Remove Index Entries

Choosing **Fix Headers and Footers** will reset the headers and footers in your manual to match the headers and footers in the attached Doc-To-Help template. It will also create odd and even headers and footers if you've chosen this in Document Preferences.

Choosing **Fix Page Setup** will match the page setup (i.e., the margins, page size and orientation) in the manual to the default in the attached template. It will also create facing pages if you've chosen it in Document Preferences.

Choosing **Remove Index Entries** will remove Word for Windows index fields and Doc-To-Help temporary index codes from the manual. If you've created extensive index entries in your manual, you may want to leave this unchecked.

If you are working with a multifile project the **Tools Repair Manual** dialog will have an additional field for rewriting RD fields. Choosing **Rewrite RD Fields** will remove any **RD** references that you used in your original main document. They will be added back in when you use **Insert New Chapter** to convert your inside files.

Reformatting the Manual

Removing Index and Table of Contents

If your original document had an index and a table of contents, you won't need these anymore. You can use the index and table of contents that was created by Doc-To-Help instead.

1. Delete the index and table of contents from the original document.
2. To update the Doc-To-Help index, place the insertion point in the index field and press **F9**. To update the Doc-To-Help table of contents place the insertion point in the table of contents field and press **F9**.

Mapping Existing Styles to Doc-To-Help Styles

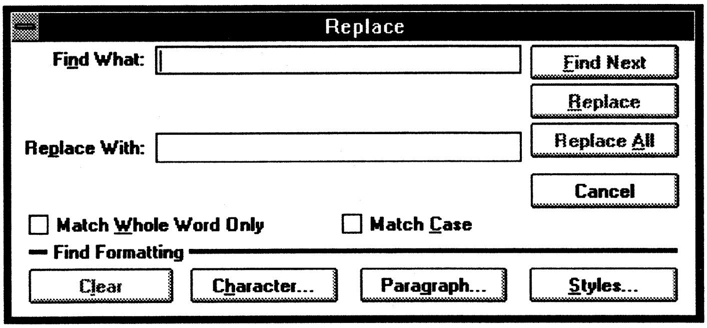
Doc-To-Help makes extensive use of heading styles in many commands (particularly when converting the manual into Help). The next step after you've run Tools Repair Manual is to apply Doc-To-Help heading styles.

If you've created your original manual in Word for Windows and you've already used the built-in heading styles, this mapping will happen automatically.

If you didn't use the built-in Heading styles in the original manual you'll need to apply the heading styles to all heading paragraphs. You could go through the new manual and apply the styles manually. However, if you want to save time, you may find it easier to use **Edit Replace** to replace the styles you've used with the heading styles.

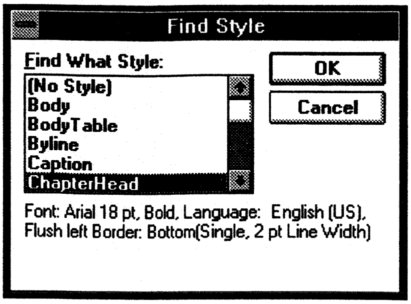
To Replace Styles Using Edit Replace

1. From the **Edit** menu choose **Replace**. The Replace dialog box will appear as shown below.



2. With the insertion point and nothing else in the **Find What** text box, click **Styles**. The **Find Style** dialog box will appear as shown below.

In this example, we've selected "ChapterHead," a style that was used in the original document. Since Doc-To-Help expects chapter headings to be formatted in Heading 1 style, we will replace the ChapterHead style with Heading 1.



Font: Arial 18 pt, Bold, Language: English (US), Flush left Border: Bottom(Single, 2 pt Line Width)

3. Select the style that you wish to replace. Click **OK**. The Replace dialog box will reappear.
4. Move the insertion point to the **Replace With** text box.
5. Click **Styles**. When the Find Styles dialog box appears, select the Doc-To-Help style you want and click **OK**.
6. Click **Replace All**. Word will apply the **Replace With** style to all paragraphs formatted with the **Find What** style. Click **OK.**

You'll probably want to repeat steps 1-6 to reformat all non-standard styles with Doc-To-Help standard styles. For example, replace Normal style with Body style, Steps style with List style, etc. If you have more than one file, consider recording a WordBasic macro as you reformat the first file. You can then run this macro on each additional file.

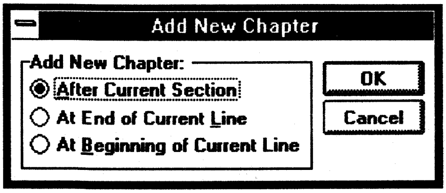
If you want more information on the Doc-To-Help styles see "Doc-To-Help Style Definitions" on page 198.

Setting Up Chapters

If you've chosen double-sided printing (i.e., different odd and even pages with each new chapter beginning on an odd page), each chapter in your document should be in a separate section. Sections are delimited by a **section break**, which appears on your screen as a double dotted line (visible when you're in **Normal** view). If you haven't separated chapters into sections in your original document, you should use insert the appropriate section break at the beginning of each chapter, as explained below.

To Insert Section Breaks Before Each Chapter

1. Position your insertion point at the beginning of the line that contains the second chapter heading. (If your document has a table of contents or title page, there will be a section break before your first chapter. If not, the first chapter will be at the beginning of the document and there is no need for a section break there.)
2. From the **Insert** menu choose **New Chapter**. The Add New Chapter dialog box will appear, as shown below.


Add New Chapter

Add New Chapter:

● After Current Section
○ At End of Current Line
○ At Beginning of Current Line

3. Choose **At Beginning of Current Line.** Doc-To-Help will go to the beginning of the current line and insert a section break. This new section will inherit the properties of the previous section.

Converting Tables

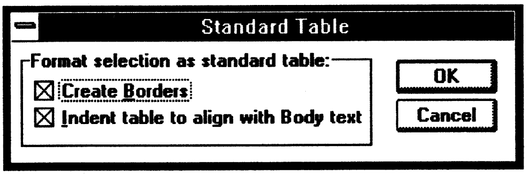
The **Standard Table** command will reformat any tables in your document to match the page setup and styles of the underlying Doc-To-Help template. Standard Table will convert both tables created with the Word for Windows table commands and tables created using tabs.

The primary advantages of using the Doc-To-Help table command over the regular Word table commands are:

- Tables will be indented to line up with body text.
- Cells will be pre-formatted in TableHeading and TableText styles.
- Borders can be applied automatically.

To Turn Text into a Standard Table

1. Highlight the table or tabbed text you want to reformat.
2. From the **Table** menu, choose **Standard Table**. The following dialog box will appear:


Standard Table

Format selection as standard table:

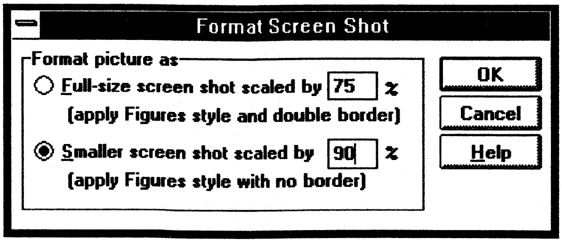
☒ Create Borders
☒ Indent table to align with Body text

3. Click **OK**.

For more information on using this feature, see “Table Menu” on page 242.

Graphics

If you had graphics in your original manual, you will need to reformat them to line up with body text in the new manual. Use the **Format Screen Shot** command to scale and format graphics in your manual. Choosing this command will display the Format Screen Shot dialog box, as shown below.



Format Screen Shot

Format picture as

○ Full-size screen shot scaled by 75 %
(apply Figures style and double border)

● Smaller screen shot scaled by 90 %
(apply Figures style with no border)

Selecting **Full-size screen shot** will scale the graphic, apply Figures style formatting and place a double border around the graphic. Choosing **Smaller screen shot** (which you should use to format screen shots of dialog boxes and smaller screen components) will scale the graphic and apply Figures style formatting. Both selections remove any frame you may have used to position your graphic.

The following scaling factors work well with Doc-To-Help templates:

Template	Screen Shot Type	Scaling Factor
D2H_NORM	Full	70%
D2H_NORM	Small (for dialog boxes, etc.)	90%
D2H_SIDE	Full	70%
D2H_SIDE	Small	90%
D2H_SMAL	Full	60%
D2H_SMAL	Small	75%

Frames

The Windows Help Compiler ignores frame formatting when you compile your document into Help. If you've used frames to position text or graphics in your original document you will need to remove the frames and reposition the formerly framed text or graphics using paragraph formatting instead. Using Format Screen Shot removes the frame around the graphic.

If you have text formatted in a frame, applying any of the Doc-To-Help styles will remove the frame around the text. If you need to make adjustments to how the paragraph or graphic is positioned on the page, you can make adjustments in **Format Paragraph**.

Vertically positioned frames are particularly tricky to work with. Once you've removed the frames, you'll have to position these paragraphs by setting the space before and after the paragraph. You can set this explicitly in the **Format Paragraph** dialog box or use the keyboard shortcuts. (For more information on the keyboard shortcuts see "Shortcut Keys" on page 203.) After you've removed the frames you may notice that the order of the paragraphs has changed. The paragraphs' vertical positioning has reverted to its true order in the document. Before you remove the frames, check the order of the framed paragraphs by making sure **Normal** is selected in the **View** menu.

Glossary of Terms

Make sure that Glossary of Terms is selected in Format Document Preferences.

If your original document contained a Glossary of Terms you can convert it to Doc-To-Help with a few simple steps. Once you're finished, you can add additional glossary terms using **Insert Glossary Term**.

To Convert a Glossary of Terms

1. Move the text of your glossary of terms to the Glossary of Terms section at the end of the document. Do not delete the Glossary of Terms heading that was created by Doc-To-Help. This heading contains a special field that Doc-To-Help uses to locate the Glossary of Terms.
2. Apply the Heading 5 style to all defined terms.
3. Apply the Definition style to all definitions.

Indexing

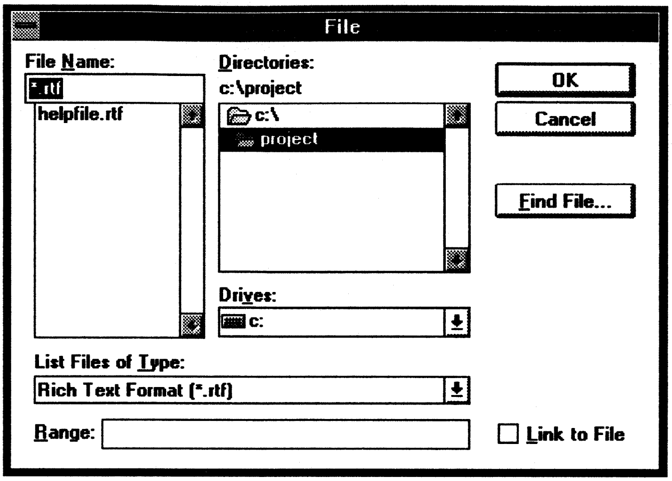
If the original document was created in Word for Windows and contains index entry fields you can leave them as is and add any additional index targets and associated tags using the commands in **Tools Indexing**.

If you have obsolete index entry fields in your document or if you want to create an index from scratch, make sure that **Remove Index Entries** is selected when you run **Tools Repair Manual**. Then, follow the steps to build an index target list.

(See "Indexing" on page 65 and "Tools Indexing" on page 229.)

Working with Pre-Existing RTF Files

You can easily import coded RTF files into your current document, and have Doc-To-Help remove unwanted formatting, footnotes, etc. The procedure is as follows: First, place the insertion bar where you want to insert the coded RTF file, then choose **Insert Coded RTF** file from the **Insert** menu. Doc-To-Help will display a message box, followed by the Insert File dialog box, shown below.



File

File Name: *.rtf

helpfile.rtf

Directories: c:\project

c:\
project

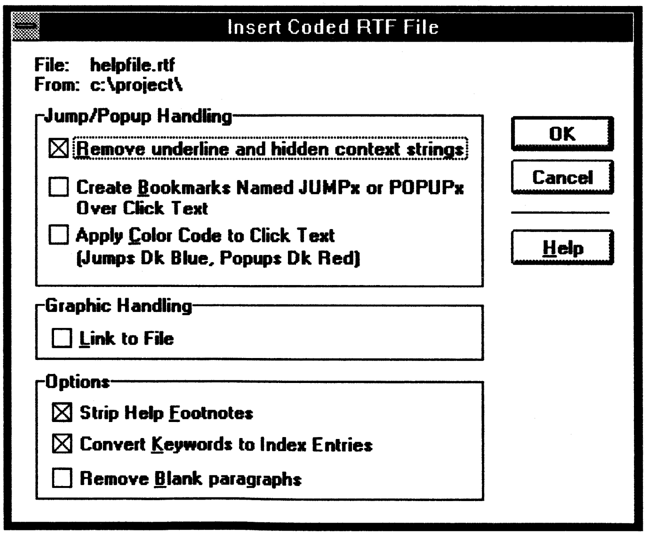
Drives: c:

List Files of Type: Rich Text Format (*.rtf)

Range:

□ Link to File

Choose the file you want to insert and click OK. Doc-To-Help will display the Insert Coded RTF File dialog box, as shown below:



Insert Coded RTF File

File: helpfile.rtf
From: c:\project\

Jump/Popup Handling

☒ Remove underline and hidden context strings

□ Create Bookmarks Named JUMPx or POPUPx Over Click Text

□ Apply Color Code to Click Text (Jumps Dk Blue, Popups Dk Red)

Graphic Handling

□ Link to File

Options

☒ Strip Help Footnotes

☒ Convert Keywords to Index Entries

□ Remove Blank paragraphs

Jump/Popup Handling

Choosing **Remove underline and hidden context strings** will make Doc-To-Help search for and remove any jump and popup formatting. Choosing **Create Bookmarks Named JUMPx and POPUPx** will mark with bookmarks those places where jumps and popups used to exist. Choosing **Apply Color Code to Click Text** will change the color of text that marked jumps and popups in the original RTF file.

Graphics Handling

When importing the coded RTF file, Doc-To-Help will attempt to replace all {bmc}, {bml} and {bmr} references with their corresponding graphics. Doc-To-Help first looks in the directory where the RTF file is saved for any graphics. If Doc-To-Help can't find the graphics there, it looks in the original project's HPJ file for the BMROOT option setting. If Doc-To-Help still cannot find the graphic, Doc-To-Help will display the Insert Picture dialog box and allow you to specify where to find the graphic in question.

Turning the **Link to File** option on will replace the graphic with a Word for Windows IMPORT field, creating a link to the disk-based graphic.

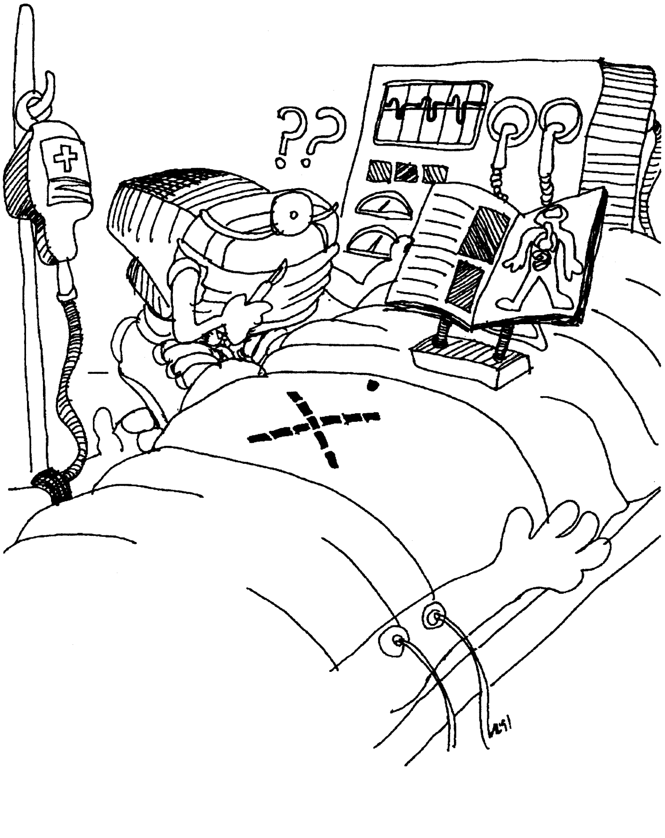
Options

Choosing **Strip Help Footnotes** will remove any help-specific footnotes (#,$,!,+, etc.). Choosing **Convert Keywords to Index Entries** will replace any keywords associated with a topic with Word for Windows XE fields. Choosing **Remove blank paragraphs** will remove any extraneous paragraphs.

Topics Will Be Formatted in Heading 2 Style

When Doc-To-Help reformats the inserted RTF file, the heading of each topic will be formatted in Heading 2 style. This will allow you to easily reorganize the newly inserted file using outline view.

Advanced Topics



Advanced Topics

Marking Text Manual/Help Only

Overview

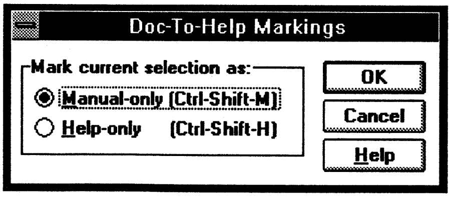
As you write documentation, you will no doubt write passages that you don't want included in the accompanying Help file. For example, the Tutorial/Basics section for Doc-To-Help includes passages and screen shots not found in the corresponding section of the online help file. Likewise, some information in the Help file can't be found in the written documentation. While you could make two separate files — one for the manual and the other for the Help — you will probably find it much easier to maintain just one source document that feeds both the manual and the online help.

You can mark text to be included and/or excluded from either the manual or the Help file using the **Format Doc-To-Help Markings** command.

To Mark Text for the Manual Only

The shortcut key for marking text for the manual only is CTRL+SHIFT+M.

1. Highlight the text and/or graphics you want to appear only in the manual.
2. From the **Format** menu, choose **Doc-To-Help Markings**. The Doc-To-Help markings dialog box will appear:



Doc-To-Help Markings

Mark current selection as:

● Manual-only (Ctrl-Shift-M)

○ Help-only (Ctrl-Shift-H)

3. Choose **Manual** and click **OK.**

The selected text will be formatted in red.

To Mark Text for Help Only

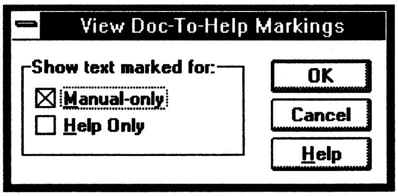
The shortcut key for marking text for help only is CTRL+SHIFT+H.

1. Highlight the text you want to appear only in online help.
2. From the **Format** menu, choose **Doc-To-Help Markings**.
3. When the dialog box appears, choose **Help** and click **OK.**

The selected text will be formatted in magenta.

Viewing Doc-To-Help Markings

The View Doc-To-Help Markings command will allow you to view (and print) text marked for the manual only, text marked for help only or both manual-only and help-only text. Choosing this command produces the following dialog box:



View Doc-To-Help Markings

Show text marked for:

☒ Manual-only

☐ Help Only

Make sure you turn off display of Help only text before you update the Table of Contents or Index, or print the file.

Choosing **Manual** will display text formatted in red and black. Choosing **Help** will display text formatted in magenta and black. Choosing both will display all manual and help text.

Note: You may notice that View Doc-To-Help markings will ignore table borders in tables marked as Help only or Manual only. To suppress display of the table borders, change the row height of the table to 1 pt using **Table Row Height** and change the border color to white using **Format Border**. Remember to change it back if you want to view the table again!

Cross References and Page References

Overview

Throughout this manual there is text that looks like this:

> **For more information, see "Advanced Techniques" on page xx.**

A good manual usually contains many cross references. In addition, Doc-To-Help automatically converts cross references into hypertext "jumps" when converting the manual into Help.

Word for Windows has a facility that allows you to enter cross references and page references that change dynamically; that is, if you change the text you want to reference or if you move that text to a different page, Word will update the reference automatically. You can produce these dynamic references either by inserting special field codes or by using Doc-To-Help's **Insert Cross Reference** command.

However, before you can use this command you will need to insert bookmarks into the document.

Note: Word for Windows does not support cross references that refer to bookmarks contained in external files. You can only refer to bookmarks contained in the file you are working on. You can, however, have hypertext links to external files with **Insert Hypertext Link**. *(See "Insert Hypertext Link" on page 210 for more information.)*

Bookmarks

As the name implies, Word's bookmark command allows you to insert place markers into your document that you can access and reference easily.

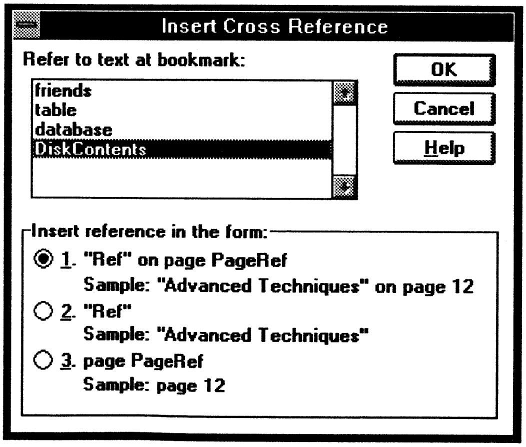
To Insert a Bookmark That Will Be Used as a Cross Reference

1. Highlight the word or phrase you want to reference. Make sure that you do NOT include a paragraph marker.
2. From the **Insert** menu, choose **Bookmark**.
3. When the dialog box appears, enter the name you want to assign to the selected text and click **OK**.

Cross References/Page References

To Insert a Cross Reference/Page Reference

1. Position your cursor where you want the reference to appear.
2. From the **Insert** menu choose **Cross Reference**. A dialog box similar to the one shown below will appear.



Insert Cross Reference

Refer to text at bookmark:

friends
table
database
DiskContents

Insert reference in the form:

● 1. "Ref" on page PageRef
Sample: "Advanced Techniques" on page 12

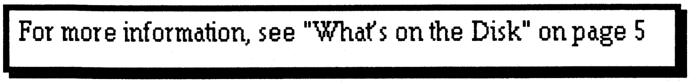
○ 2. "Ref"
Sample: "Advanced Techniques"

○ 3. page PageRef
Sample: page 12

In this example the bookmark "DiskContents" refers to the text "What's on the Disk."

3. Select the bookmark you want to reference and the type of reference you want to insert.

4. Click **OK**. Word will insert a cross reference/page reference like the one shown below.

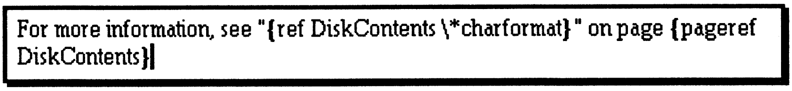


For more information, see "What's on the Disk" on page 5

These references will be updated when you print the document. You can also update the references yourself using the Update Fields command (function key **F9**).

Looking Under the Hood—Examining Ref and Pageref Field Codes

If you turn field codes on (by selecting Field Codes from the View menu) you will see text similar to the following:



```
For more information, see "{ref DiskContents \*charformat}" on page {pageref
DiskContents}
```

As you can see, the Insert Cross Reference command places special versions of the Ref and Pageref fields into your document.

Make sure bookmarks referred to by cross references do NOT include paragraph markings as the cross references may inherit the paragraph properties of the paragraph that contains the bookmark.

Inserting Hypertext Links

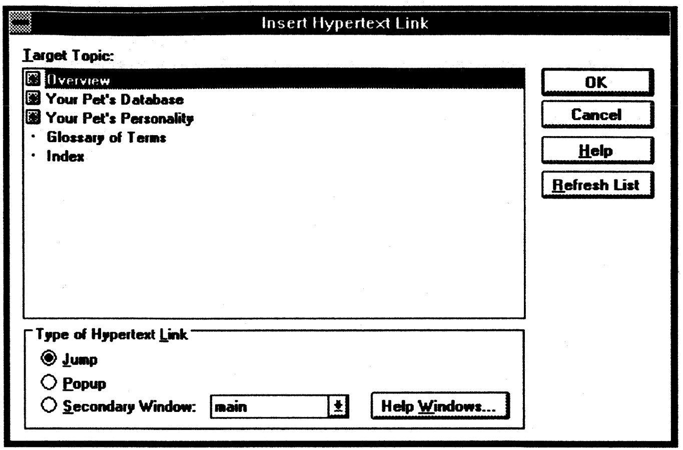
Overview

When converting a manual into Help, Doc-To-Help will automatically create hypertext jumps when if finds Ref and Pageref fields. In addition, Doc-To-Help will make any text formatted in Heading 3 or Heading 4 a subtopic of the nearest preceding text formatted in the next highest heading style. (See "How Doc-To-Help Turns Manuals into Online Help" on page 47 for more information.)

There will probably be places where you want a cross reference jump to appear in the help file but don't want it to appear in the manual. You can create these "help only" jumps and popups using the **Insert Hypertext Link** command.

To Insert a Hypertext Link

1. Select the text or graphic that you want to make into a link.
2. From the **Insert** menu choose **Hypertext Link**. The Insert Hypertext Link dialog box will appear (an example is shown below).



● Jump
○ Popup
○ Secondary Window: main

Insert Hypertext Link dialog box showing unexpanded Heading 1 topics

3. Double-click on an unexpanded topic title (a topic with a plus sign next to it) to see the subtopics associated with that topic. If you are using the keyboard, select the topic using the arrow keys and press the plus sign key (+) to expand the topic.
4. Select the topic you want to link to.
5. Choose the type of hypertext link you want to make. The link types are described in the table below.

Choosing this type of link	**Produces this result**
Jump	Moves to a different topic within the help file.
Popup	Displays a popup window on top of the current help topic.
Secondary Window	Moves to a different topic within the help file, and displays that topic in a secondary window. (For information on secondary windows, see "Help Windows" on page 223.)

If you choose either popup or secondary window, Doc-To-Help will ask you if you want to designate the target topic as an Auxiliary Help Topic.

If you choose Yes, the only way the users of your help file will be able to access this popup or secondary window jump is through user-defined hypertext links (like the one you are now creating). This means that users will not be able to access the topic from the contents topic, a browse sequence or the keyword search facility. You can change how a topic can be accessed using the Edit Help Topic Status command which is discussed below.

6. Click **OK**. Doc-To-Help will insert a special Relate field which will be turned into a jump when you make the manual into help. If you want, you can examine this field by turning field codes on. (For more information on Relate fields, see "Relate Field Syntax" on page 179.)

Note: Unlike the Insert Cross Reference command, the Insert Hypertext Link command can create links to other files.

Changing the Help Topic Status

Overview

When Doc-To-Help converts headings into help topics, each help topic is accessible though any or all of the following features of the Microsoft Windows online help system:

- The contents topic
- As a sub-topic of a main topic (which Doc-To-Help creates automatically based on the heading structure of your document)
- As a topic within a browse sequence
- Keyword search

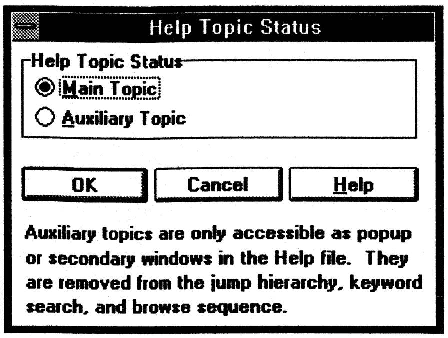
There may, however, be cases when you want a topic to be accessible only through a link created using the Insert Hypertext Link command. For example, a very brief, elliptical popup definition might be just enough when accessed in context, but too slight to stand on its own as an independent help topic. In this instance, the topic should be an **auxiliary topic**.

Even if the target topic of the link is a regular jump, you make that target topic an auxiliary topic using the Edit Help Topic Status command.

When you indicate that a target of a hypertext link will appear as either a popup definition or in a secondary window, Doc-To-Help gives you the opportunity to make the target topic an auxiliary topic. An auxiliary topic is a topic that can only be accessed though links created using the Insert Hypertext Link command. If, after making a topic into an auxiliary topic, you decide you want to make the topic into a main topic, you can do so using the Edit Help Topic Status command.

To Check and Change the Status of a Topic

1. Click in the heading of the topic you want to change.
2. From the **Edit** menu, choose **Help Topic Status**. The Help Topic Status dialog box will appear, as shown below.



Help Topic Status

● Main Topic

○ Auxiliary Topic

Auxiliary topics are only accessible as popup or secondary windows in the Help file. They are removed from the jump hierarchy, keyword search, and browse sequence.

3. If you are happy with the topic status, click Cancel; otherwise, change the topic status and click OK.

Changing the Status for More than One Topic at a Time

You can change the topic status for a group of headings by selecting these headings before you access the Edit Help Topic Status command. For example, you may want to make sure that every topic in a section is a main topic.

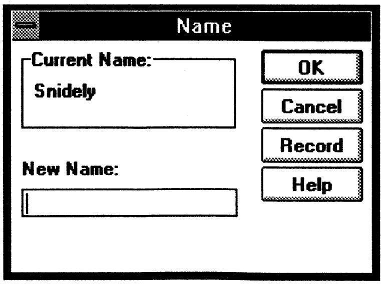
To Change the Status of a Group of Topics

1. Select all the headings you want to change (this means selecting the text between the headings, too).
2. From the **Edit** menu, choose **Help Topic Status**. The Help Topic Status dialog box will appear.
3. Choose the desired option (either Main or Auxiliary) and click **OK**.

Context-Sensitive Help

Mappings

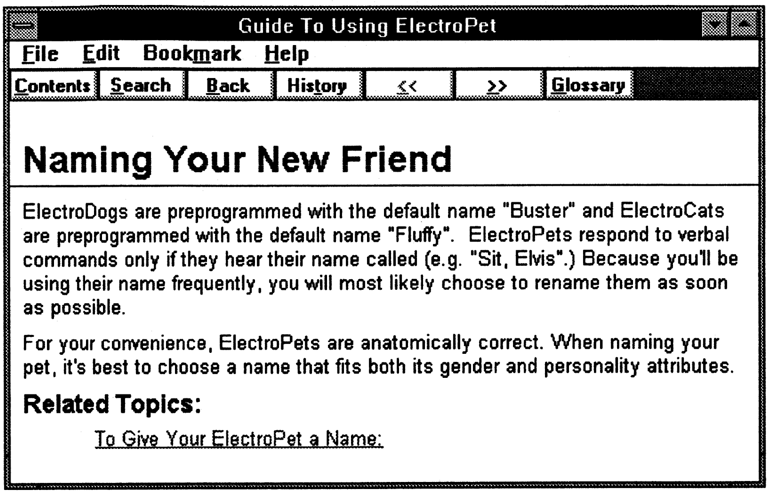
A Help file consists of a series of screens called **Help Topics**. The Help compiler assigns a context number to each context string so it can jump to the correct topic. A macro or program can be written to call a specific help topic. Consider the sample dialog box below:



Current Name: Snidely

New Name:

With context-sensitive help, the help screen for “Naming Your New Friend” will appear when the user clicks on the help button in this dialog box. That way the user doesn’t have to wade through the entire Help file to find the context-sensitive help topic as shown below.



Naming Your New Friend

ElectroDogs are preprogrammed with the default name "Buster" and ElectroCats are preprogrammed with the default name "Fluffy". ElectroPets respond to verbal commands only if they hear their name called (e.g. "Sit, Elvis".) Because you'll be using their name frequently, you will most likely choose to rename them as soon as possible.

For your convenience, ElectroPets are anatomically correct. When naming your pet, it's best to choose a name that fits both its gender and personality attributes.

Related Topics:

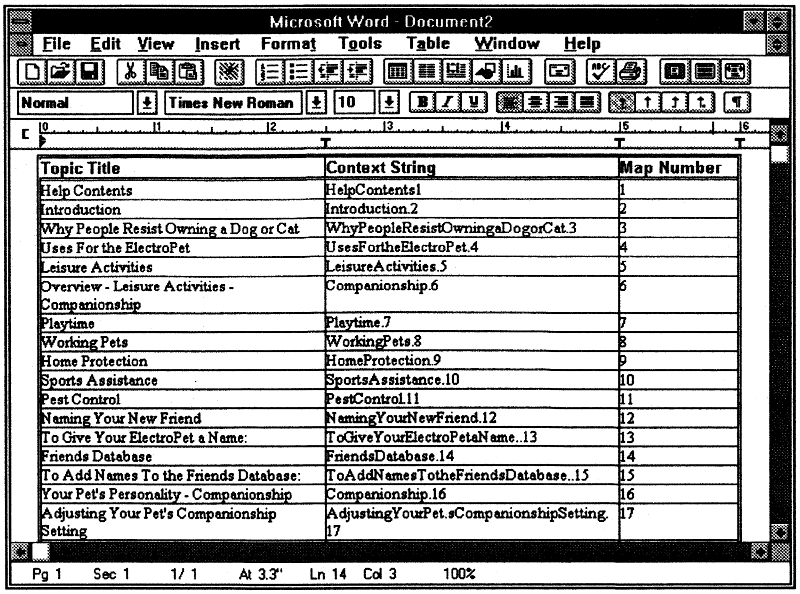
To Give Your ElectroPet a Name:

To create context-sensitive help, programmers need to know the relationship (known as a “mapping”) between your Help file’s context strings and context numbers. This mapping is created by Doc-To-Help when it writes the Help project file. You can view this mapping using the **Tools Show Context Mapping** command.

To View the Context Mapping

1. Make sure the **RTF** file is open.
2. From the **Tools** menu, choose **Show Context Mapping**.

Doc-To-Help will create a new document that lists the Topic Title, Context String and Map Number, as shown below.


Microsoft Word - Document2
File Edit View Insert Format Tools Table Window Help
Normal
Times New Roman
10
Pg 1 Sec 1 1/ 1 At 3.3" Ln 14 Col 3 100%

Topic Title	Context String	Map Number
Help Contents	HelpContents1	1
Introduction	Introduction.2	2
Why People Resist Owning a Dog or Cat	WhyPeopleResistOwningaDogorCat.3	3
Uses For the ElectroPet	UsesFortheElectroPet.4	4
Leisure Activities	LeisureActivities.5	5
Overview - Leisure Activities - Companionship	Companionship.6	6
Playtime	Playtime.7	7
Working Pets	WorkingPets.8	8
Home Protection	HomeProtection.9	9
Sports Assistance	SportsAssistance.10	10
Pest Control	PestControl.11	11
Naming Your New Friend	NamingYourNewFriend.12	12
To Give Your ElectroPet a Name:	ToGiveYourElectroPetaName..13	13
Friends Database	FriendsDatabase.14	14
To Add Names To the Friends Database:	ToAddNamesTotheFriendsDatabase..15	15
Your Pet's Personality - Companionship	Companionship.16	16
Adjusting Your Pet's Companionship Setting	AdjustingYourPet.sCompanionshipSetting.17	17

This command can only be run after Doc-To-Help has written the Help project file. The Help project file is written when you choose the **Tools Compile** command and has the same name as your master document with the extension HPJ.

You can give a printout of the context mapping with the associated Help Project file to the programmer who will be using the Help file. See "Appendix: Accessing Help" on page 245 for examples of how WordBasic, Excel macros, VisualBasic and C use context mapping to call context-sensitive help.

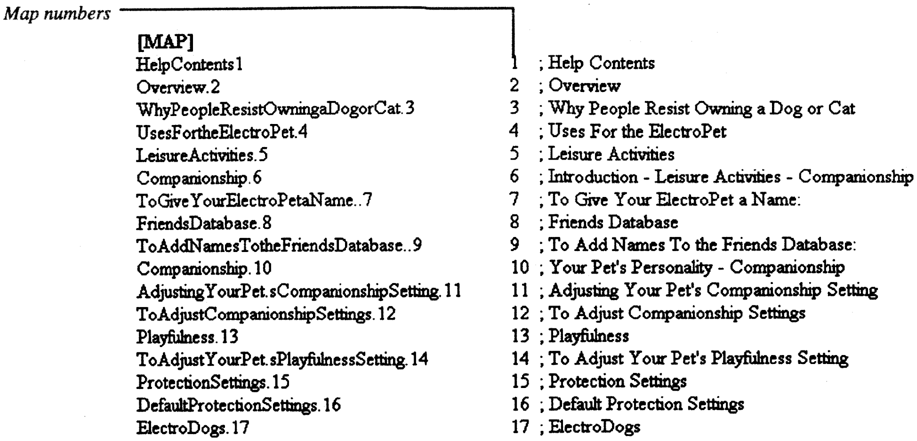
Using Existing Context Strings and Map Numbers

While Doc-To-Help insists on creating context strings and map numbers for you when it initially converts a manual into help, you can use your own map numbers if you want to. This means that if map numbers have already been coded into the application for which you are developing context-sensitive help, that application does not have to be recompiled using new map numbers.

To Use Existing Map Numbers

1. Get a list of map numbers from the developer of the application.
2. Use Doc-To-Help to convert your manual into an RTF format and to write the help project file (HPJ file). Do not compile the file into help.
3. Using a text editor that doesn't convert tabs into blank spaces, open the HPJ file. An example is shown below.

Note: Avoid using DOS EDIT as this editor will convert tabs into blank spaces. We recommend using Word for Windows to edit HPJ files.


Map numbers

```
[MAP]
HelpContents1                               1  ; Help Contents
Overview.2                                  2  ; Overview
WhyPeopleResistOwningaDogorCat.3            3  ; Why People Resist Owning a Dog or Cat
UsesFortheElectroPet.4                      4  ; Uses For the ElectroPet
LeisureActivities.5                         5  ; Leisure Activities
Companionship.6                             6  ; Introduction - Leisure Activities - Companionship
ToGiveYourElectroPetaName..7                7  ; To Give Your ElectroPet a Name:
FriendsDatabase.8                           8  ; Friends Database
ToAddNamesTotheFriendsDatabase..9           9  ; To Add Names To the Friends Database:
Companionship.10                           10  ; Your Pet's Personality - Companionship
AdjustingYourPet.sCompanionshipSetting.11  11  ; Adjusting Your Pet's Companionship Setting
ToAdjustCompanionshipSettings.12           12  ; To Adjust Companionship Settings
Playfulness.13                             13  ; Playfulness
ToAdjustYourPet.sPlayfulnessSetting.14     14  ; To Adjust Your Pet's Playfulness Setting
ProtectionSettings.15                      15  ; Protection Settings
DefaultProtectionSettings.16               16  ; Default Protection Settings
ElectroDogs.17                             17  ; ElectroDogs
```

4. Change the map numbers as needed.
5. Save and close the HPJ file.
6. From the **Tools** menu, choose **Compile**.

The next time you convert your document into a help file, Doc-To-Help will examine the entries in the pre-existing HPJ file and respect your custom map number entries. If, however, you change the heading text in a topic, Doc-To-Help will think this is a new topic (as opposed to the same topic with a different heading) and insist on generating a new map number. If this happens, edit the HPJ file as described above and make any necessary changes.

Using Hexadecimal Map Numbers

When Doc-To-Help updates the [MAP] section of the HPJ file, it examines the entries in the pre-existing HPJ file and respects your custom map numbers. Doc-To-Help can read hexadecimal map numbers that contain a 0x prefix but will, by default, write these entries back into the HPJ file as decimal numbers. You can make Doc-To-Help preserve these numbers as hexadecimal entries if you add the line

MapHex=1

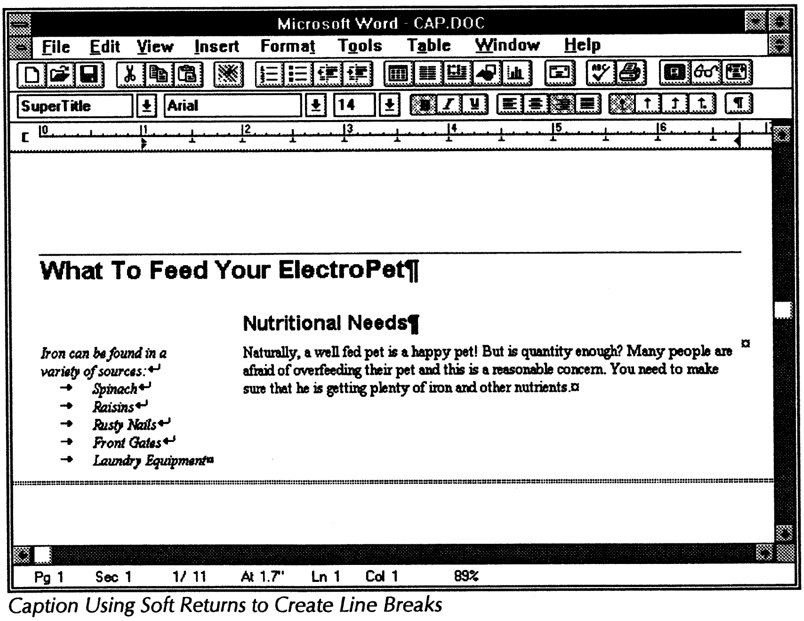
in the [Make Into Help] section of the DOC2HELP.INI file.

Lists and Multiple Paragraphs in Captions

Doc-To-Help converts captions in the manual to popup definitions in the Help file. With the **Link to Caption** command you can link text in the adjoining paragraph with a paragraph in the caption cell. When the document is converted to Help, the text linked to the caption becomes a definition button and when clicked will display the text of the caption it's linked to.

Doc-To-Help lets you have more than one caption in a single cell as long as each caption is in a separate paragraph. You select which caption you want to link to by placing the insertion point in the desired paragraph and then selecting **Link to Caption**.

A problem arises when you want to have a single caption that contains line breaks. Separating the lines into paragraphs won't work because Doc-To-Help will see each paragraph as a separate caption in the **Link to Caption** command. The easy way to solve this problem is to use line breaks (**SHIFT+ENTER**) instead of hard returns at the end of each paragraph. If you want to indent the line you'll need to use tabs. Because the caption is in a table cell you'll need to press **CTRL+TAB** to enter a tab. In the following example soft returns were used to create line breaks in the captions.


Microsoft Word - CAP.DOC
File Edit View Insert Format Tools Table Window Help
SuperTitle
Arial
14
Pg 1 Sec 1 1/ 11 At 1.7" Ln 1 Col 1 89%

What To Feed Your ElectroPet¶

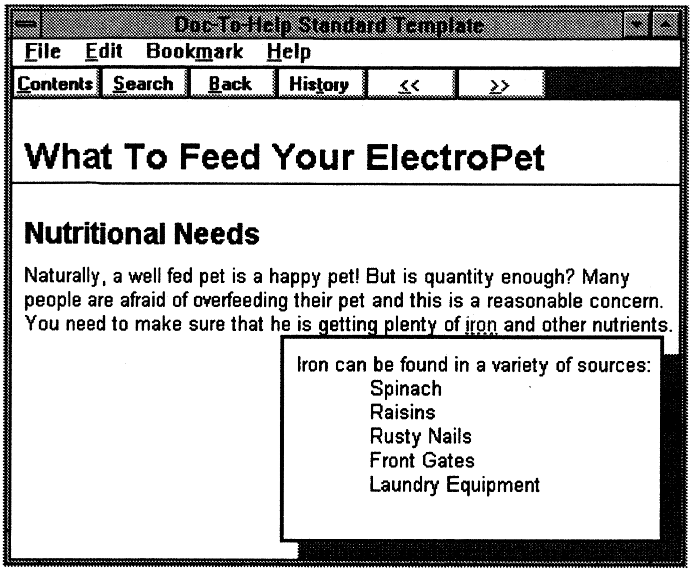
Iron can be found in a variety of sources:↵
→ *Spinach↵*
→ *Raisins↵*
→ *Rusty Nails↵*
→ *Front Gates↵*
→ *Laundry Equipment¤*

Nutritional Needs¶

Naturally, a well fed pet is a happy pet! But is quantity enough? Many people are afraid of overfeeding their pet and this is a reasonable concern. You need to make sure that he is getting plenty of iron and other nutrients.¤

Caption Using Soft Returns to Create Line Breaks

Doc-To-Help converts this page to the following Help screen:


Doc-To-Help Standard Template
File Edit Bookmark Help
Contents Search Back History << >>

What To Feed Your ElectroPet

Nutritional Needs

Naturally, a well fed pet is a happy pet! But is quantity enough? Many people are afraid of overfeeding their pet and this is a reasonable concern. You need to make sure that he is getting plenty of iron and other nutrients.

Iron can be found in a variety of sources:
Spinach
Raisins
Rusty Nails
Front Gates
Laundry Equipment

Multiline Popup Definition

Changing the Appearance of the Manual or Help File

Word for Windows gives you an enormous degree of flexibility in formatting your documents and, if you so desire, the Doc-To-Help templates to which your documents are attached.

The key to these capabilities is Word's ability to let you change and add styles in the documents and templates. While a tutorial on how to define, merge, and apply styles is beyond the scope of this guide, it is highly recommended that you read the pertinent sections in the Word for Windows manual if you're not comfortable working with styles.

Redefining Styles in the Manual

If you want to change the appearance of your printed manual, redefining styles in the documents is often the fastest way. However, there are a few points to bear in mind if you do change the style definitions in your documents:

- Redefining styles in the document will not affect other manuals you plan to write in the future. To make these changes "stick" to subsequent projects, you'll have to redefine the styles in the Doc-To-Help manual templates themselves.
- If you do decide to redefine styles in the Doc-To-Help templates, be sure to maintain consistency in the style definitions of the master template you're changing, and its associated inside chapter template. That is, if you change styles in D2H_NORM

or D2H_SMAL make the same changes in D2H_IN or D2H_INSM. (See "Help Templates" on page 195.)

Word's **Style Merge** *feature can help maintain consistency in your documents.*

- If you're working with a multifile project and you decide to modify the styles at the document level, be sure to maintain consistency in each of the documents comprising your manual. In particular, the **toc** and **index** styles must be defined the same in each of your documents, or the Table of Contents and Index in the Master Document will be formatted inconsistently.
- The left indent in Body and other styles is designed to allow room for captions in the margin. If you plan to use this feature of Doc-To-Help, it is recommended that you do not change this indent.
- Formatting changes in the manual that are style-based, will not translate into Help. This is because the Help file takes its formatting instructions from a different template called D2H_HELP.

Redefining Styles in Help

When Doc-To-Help makes the manual into Help it applies the styles stored in D2H_HELP to the RTF file it creates. While you're free to format and edit your project's RTF files after they have been reformatted as Help, these changes will be overwritten the next time you make your manual into Help. If you need to make formatting changes to the Help file that will stay in effect after subsequent revisions, modify the styles in the Help Template (D2H_HELP).

Adding New Styles

Most of the styles in D2H_NORM have a two-inch indent.

Feel free to add your own styles to the Doc-To-Help templates. If you do create your own styles for the manual, make sure you create a corresponding style for the help file. For example, let's say you create a style in your manual called SpecialBullets that indents text two inches from the left. When you make the manual into help, Doc-To-Help will respect the style name and preserve the style in the help file, leaving the two-inch indent. The solution is to also create a style called SpecialBullets in D2H_HELP.DOT that is the same as the manual's version, but without the two-inch indent.

Changing the Color of Jump Text

The default color of jump text for all Help files is green. You (and your users) can change the color by modifying the WIN.INI file. Simply add the following section to WIN.INI:

```
[Windows Help]
Jumpcolor=255 0 0
Popupcolor=255 0 0
```

The values for Jumpcolor and Popupcolor represent red, green and blue respectively. In the example above the jump and popup colors have been turned to red (0 0 255 would be blue, etc.). You may want to examine the [colors] section of the WIN.INI file to see how these three numbers are used to control the colors of your scroll bars, menu bars, push buttons, and so on.

Hard-Coding the "Look" of Jump Text

While we recommend leaving the default colors alone and letting the user determine the jump colors by manipulating the WIN.INI file as described in "Changing the Color of Jump Text" above, you can "hard-code" the color of jump text, and eliminate the underscore.

Hard-coding the jump text colors requires editing the RTF file. With the RTF file open, make sure hidden text is showing and search for the jump text that you want to change (this text will be formatted with a single underline for popups and a double underline for jumps). Between the underlined text and the hidden text which immediately follows, insert one of the following special characters in hidden text between the jump text and the hidden string:

This Character	Produces This Result
%	no visible underscore
*	visible underscore

Use the Format Character Color command to color the jump text as desired.

Note: WexTech has posted a macro on CompuServe the automates this process. Type GO WEXTECH at any CompuServe prompt.

Symbols and Special Characters

Since Windows 3.1 and the current versions of the Help Compiler and Help Engine (HC31.EXE and HCP.EXE) support the Symbol font, Doc-To-Help employs a scheme for mapping Symbol fields. If a Symbol field specifies the Symbol font, the field will be replaced by a formatted character which will be recognized by the Help Compiler.

Symbol fields which specify an unsupported font (Wingdings, etc.) will be left alone, and consequently ignored by the Help Compiler (these appear as a blank space in the Help file). The exception to this is that if the unsupported Symbol field appears at the beginning of a paragraph and is followed by a tab, it will be converted to a standard Symbol font bullet.

Em dashes in other fonts will be replaced with their Symbol font equivalent. En dashes will be converted to hyphens, since there is no en dash character in the Symbol font.

The **Tools Doc-To-Help Diagnostics** command gives you the option to check for unsupported symbols. You should run this command before making the manual into Help. (See "Tools Doc-To-Help Diagnostics" on page 236 for more information.)

Tables and Borders

As we mentioned in the tutorial, The Doc-To-Help Table Standard Table command enables you to both insert and format tables more easily than with the regular Word for Windows table command. Using this command you can create tables that are indented to line up with body text and that have borders applied automatically.

Unfortunately, the Microsoft Windows Help Compiler has a lot of trouble with borders. In fact, the help compiler is pretty much incapable of assimilating any table border formatting and produces tables in your help file that do not have any borders.

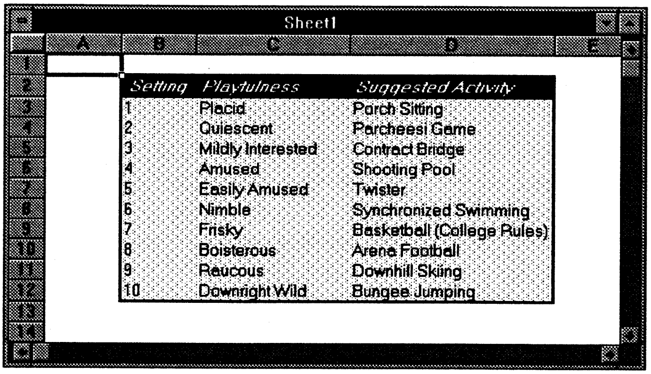
There are, however, several techniques you can employ for producing very attractive tables in both your manual and your help file.

The first techniques, creating a table in Microsoft Excel and copying into Word for Windows, requires the most effort but produces the best results. The second technique, inserting a graphic border under the table headings, produces an attractive table and requires less effort than formatting a table in Excel. The third method, applying a border to just the heading text in a cell as opposed to the entire cell, requires the least effort but produces a border that has gaps in it.

Creating Stylish Tables Using Microsoft Excel

If you want to produce a fancy table with lots of borders and shading, create the table in a a drawing or spreadsheet package and copy the table into Word for Windows as a picture. This table will look good in both the printed and online document.

1. Create and format a table using Microsoft Excel. An example is shown below.



Setting	Playfulness	Suggested Activity
1	Placid	Porch Sitting
2	Quiescent	Parcheesi Game
3	Mildly Interested	Contract Bridge
4	Amused	Shooting Pool
5	Easily Amused	Twister
6	Nimble	Synchronized Swimming
7	Frisky	Basketball (College Rules)
8	Boisterous	Arena Football
9	Raucous	Downhill Skiing
10	Downright Wild	Bungee Jumping

2. Select all the cells in the table.
3. While holding down the SHIFT key, choose **Copy Picture** from the **Edit** menu.
4. When the dialog box appears, choose **Display as Printed** and **Picture** from the applicable group boxes.
5. Click **OK**.
6. Activate Microsoft Word.
7. Position your cursor where you want the table to appear and choose **Paste** from the **Edit** menu.

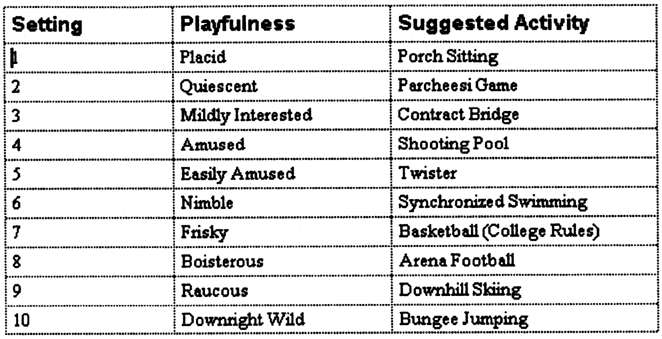
Creating a Table with a Single Continuous Border

The steps below describe how you can create a table that contains a single continuous border. This table will look good in both the printed and online document.

Remember, the Standard Table command will behave differently depending on whether your cursor is in a table and whether any text is selected. (For more information, see "Table Standard Table" on page 242.)

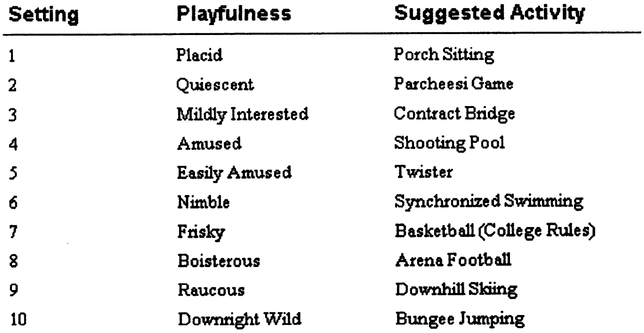
1. Use the Doc-To-Help **Table Standard Table** to create a new table or to format an existing table. When the Standard Table dialog box appears, make sure Create Borders is *not* selected.

   A sample table is shown below.



Setting	Playfulness	Suggested Activity
1	Placid	Porch Sitting
2	Quiescent	Parcheesi Game
3	Mildly Interested	Contract Bridge
4	Amused	Shooting Pool
5	Easily Amused	Twister
6	Nimble	Synchronized Swimming
7	Frisky	Basketball (College Rules)
8	Boisterous	Arena Football
9	Raucous	Downhill Skiing
10	Downright Wild	Bungee Jumping

2. Click in the first cell in the second row of the table, immediately to the left of any text.
3. Press CTRL+SHIFT+ENTER. This will place a blank line between the table heading and the rest of the table.
4. Type **TB** and press function key F3.
5. Press the DELETE key. An example of a formatted table (with gridlines turned off) is shown below.



Setting	Playfulness	Suggested Activity
1	Placid	Porch Sitting
2	Quiescent	Parcheesi Game
3	Mildly Interested	Contract Bridge
4	Amused	Shooting Pool
5	Easily Amused	Twister
6	Nimble	Synchronized Swimming
7	Frisky	Basketball (College Rules)
8	Boisterous	Arena Football
9	Raucous	Downhill Skiing
10	Downright Wild	Bungee Jumping

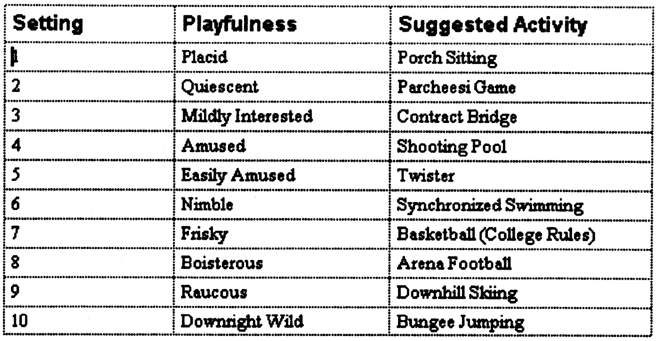
Creating a Table with a Broken Border

The steps below describe how you can create a table that contains a single broken border.

Remember, the Standard Table command will behave differently depending on whether your cursor is in a table and whether any text is selected. (For more information, see "Table Standard Table" on page 242.)

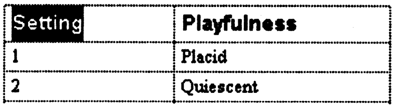
1. Use the Doc-To-Help **Table Standard Table** to create a new table or to format an existing table. When the Standard Table dialog box appears, make sure Create Borders is *not* selected.

   A sample table is shown below.



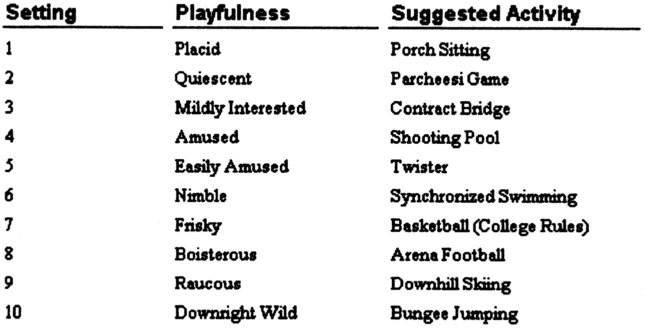
Setting	Playfulness	Suggested Activity
1	Placid	Porch Sitting
2	Quiescent	Parcheesi Game
3	Mildly Interested	Contract Bridge
4	Amused	Shooting Pool
5	Easily Amused	Twister
6	Nimble	Synchronized Swimming
7	Frisky	Basketball (College Rules)
8	Boisterous	Arena Football
9	Raucous	Downhill Skiing
10	Downright Wild	Bungee Jumping

2. Highlight just the text in the first cell of the table, not the entire cell, as shown below.



Setting	Playfulness
1	Placid
2	Quiescent

3. From the Format menu, choose Border.
4. When the Border Paragraphs dialog box appear, choose a single bottom border and click OK.
5. Repeat steps 2-4 for the remaining headings in the table. An example of a formatted table (with gridlines turned off) is shown below.



Setting	Playfulness	Suggested Activity
1	Placid	Porch Sitting
2	Quiescent	Parcheesi Game
3	Mildly Interested	Contract Bridge
4	Amused	Shooting Pool
5	Easily Amused	Twister
6	Nimble	Synchronized Swimming
7	Frisky	Basketball (College Rules)
8	Boisterous	Arena Football
9	Raucous	Downhill Skiing
10	Downright Wild	Bungee Jumping

Graphics in the Help File

Formats Supported

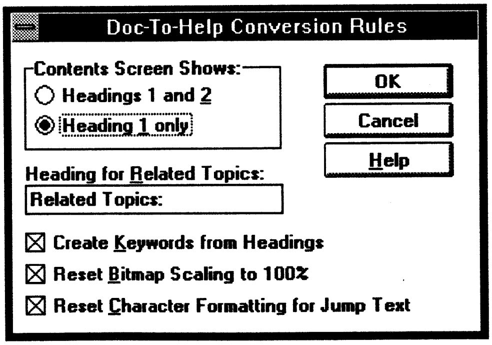
Doc-To-Help supports monochrome and 16-color bitmaps (.BMP), Paintbrush (.PCX), Tagged Image Format (.TIF) ,Windows Metafiles (.WMF) and virtually any graphic that can be inserted into a Word for Windows file. While you can use PCX and TIF files in the document, the results when making into Help are not always satisfactory. You may wish to convert graphics in these or other formats to bitmaps (.BMP) or metafiles (.WMF) before you paste them in your Help file. Refer to the documentation for your graphics program for information on saving your graphics as BMP or WMF.

Resetting Scaled Graphics

Graphics can be scaled to whatever size you desire in the manual, using the standard Word for Windows scaling commands. These graphics will usually print very nicely, depending on your printer, but the screen rendering of scaled bitmaps is generally poor. For this reason, Doc-To-Help gives you the option to reset scaled graphics to their original size in the Help file. This setting is available in the **Doc-To-Help Options** dialog box and is checked by default.

To Reset Scaled Graphics to 100% in the Help File

1. From the **Format** menu choose **Set Project Options**. When the Set Project Options dialog box appears, click on **Help Options>>**.
2. Click on **Conversion Rules**. The Conversion Rules dialog box will appear, as shown below.



Contents Screen Shows:
○ Headings 1 and 2
● Heading 1 only

Heading for Related Topics: Related Topics:

☒ Create Keywords from Headings
☒ Reset Bitmap Scaling to 100%
☒ Reset Character Formatting for Jump Text

3. Make sure **Reset Bitmap Scaling to 100%** is checked and click **OK**.

Note: You can also access the Conversion Rules dialog box from the Reformat as Help File dialog box.

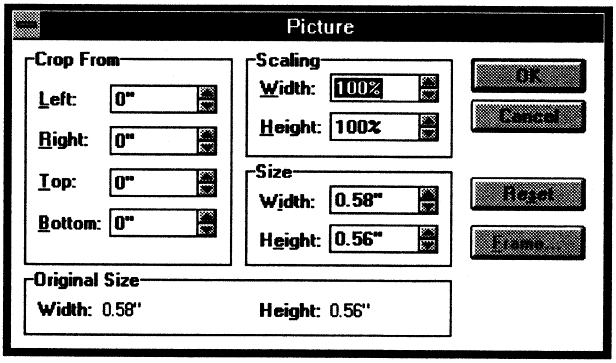
If the Bitmap Is Too Big for the Help File

If you didn't install MS Draw along with Word for Windows, you'll need to run Word Setup to install it.

If the bitmap is large it may take up too much room in the help screen. You should consider rescaling it in another graphics program rather than using the Word for Windows scaling commands. Certain graphics applications might do a better job of scaling bitmaps, particularly those which can create a vector graphic by tracing the bitmap. You can use this type of feature to create a "pre-scaled" graphic to insert into your Doc-To-Help manual. However, if you must have scaled bitmaps in your Help file and you don't have a sophisticated graphics program, you can use the MS Draw application included with Word for Windows to scale the graphic.

To Scale Graphics Using Draw

1. If the bitmap hasn't been inserted into your manual, use the **Insert Picture** command to place it into the document.
2. If the bitmap is already scaled in your document, reset scaling to 100% as follows:
   a. Select the bitmap.
   b. From the **Format** menu, choose **Picture**.
   c. Set the **Width** and **Height** options in the **Scaling** group to 100% as shown below.



Picture

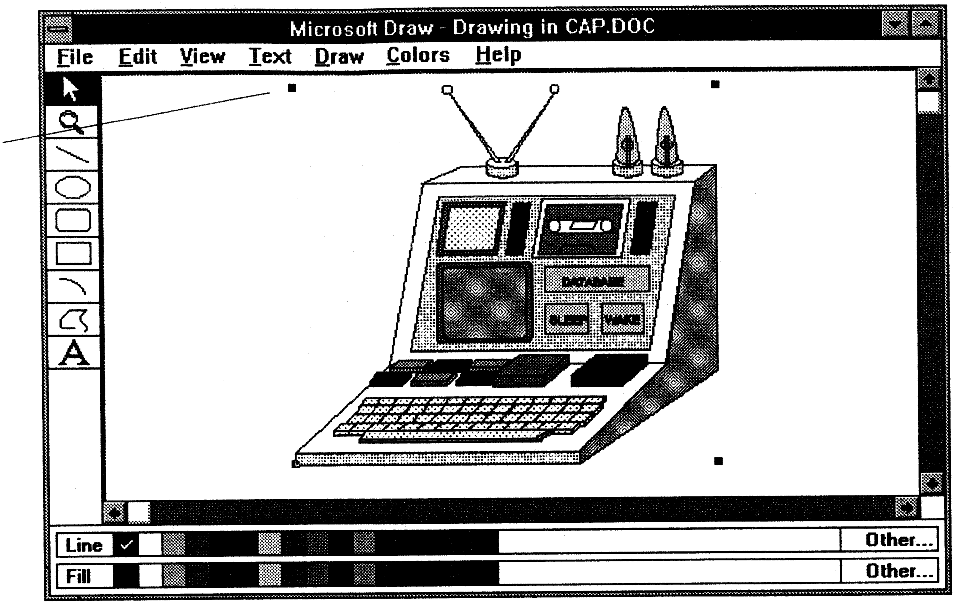
Crop From
Left: 0"
Right: 0"
Top: 0"
Bottom: 0"

Scaling
Width: 100%
Height: 100%

Size
Width: 0.58"
Height: 0.56"

Original Size
Width: 0.58"   Height: 0.56"

   d. Click **OK**.
3. Double-click on the bitmap to start the **MS Draw** application.
4. Drag the **Resize Handles** to visually scale the bitmap to whatever size you want.

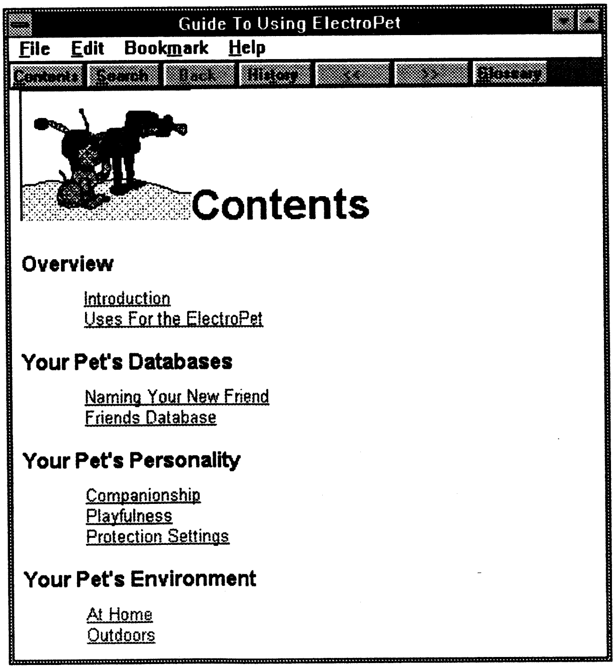

Microsoft Draw - Drawing in CAP.DOC
File Edit View Text Draw Colors Help
DATABASE
SLEEP
WAKE
Line ✓
Other...
Fill
Other...

Resize Handles are the squares at the corners of the graphic.

5. From the **File** menu, choose **Exit**. When you see the dialog box asking if you want to update the source document, answer **Yes**.

Placing a Graphic on the Contents Topic

If you want, you can add a graphic to your help file's contents topic, as shown below.


Guide To Using ElectroPet
File Edit Bookmark Help
Contents Search Back History << >> Glossary

Contents

Overview

Introduction
Uses For the ElectroPet

Your Pet's Databases

Naming Your New Friend
Friends Database

Your Pet's Personality

Companionship
Playfulness
Protection Settings

Your Pet's Environment

At Home
Outdoors

To Place a Graphic on the Contents Topic

1. If your manual doesn't have a Table of Contents, insert the Table of Contents using the **Format Set Project Options** command.
2. Place the insertion point at the beginning of the Contents Title.
3. Insert the graphic at the insertion point either by pasting from the clipboard or by using the **Insert Picture** command.
4. Mark the graphic for help only by highlighting the graphic and pressing **CTRL+SHIFT+H**.
5. If you want, press **SHIFT+ENTER** to place a line break between the graphic and the Contents Title.

Making a Graphic into a Hot Spot

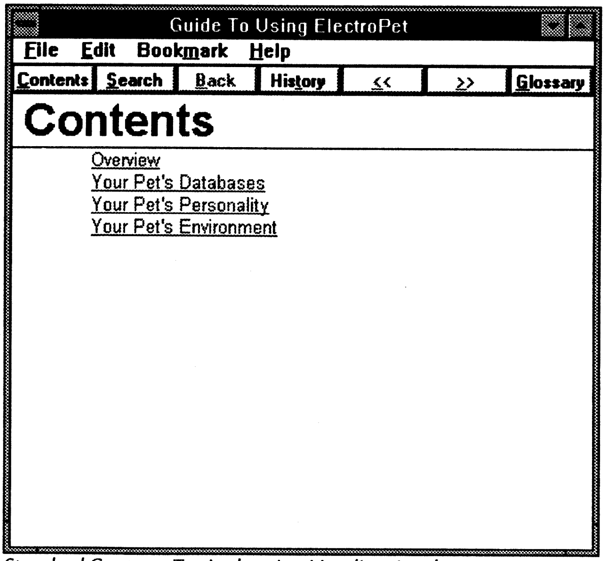
It's easy to make a graphic into a hot spot. Just select the graphic, and choose **Hypertext Link** from the **Insert** menu.

For more information, see "Insert Hypertext Link" on page 210.

Creating a "Stylish" Contents Screen

Adding Buttons to the Contents Topic

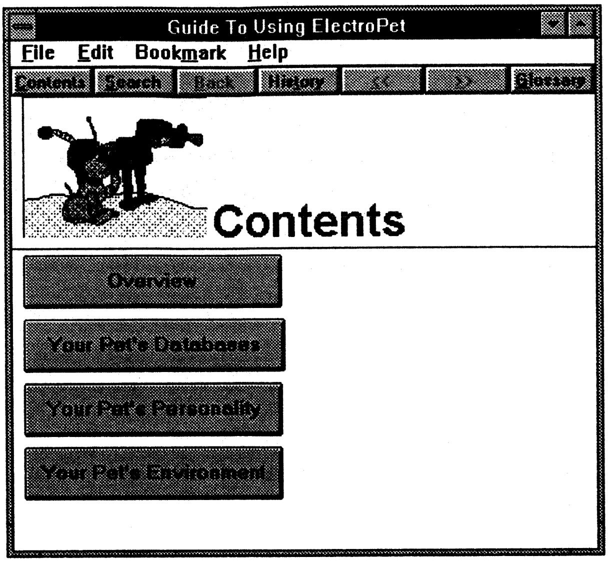
With very little effort you can take a screen that looks like this:



Contents

Overview
Your Pet's Databases
Your Pet's Personality
Your Pet's Environment

Standard Contents Topic showing Heading 1 only and make it look like this:



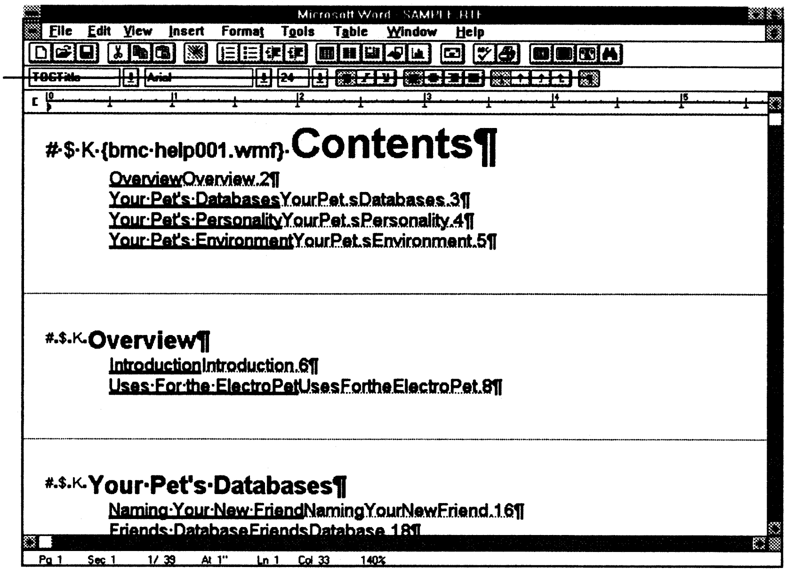
Stylish Contents Topic

While you can certainly use the Insert Hypertext Link command to turn graphics into buttons within the body of your help file, here we'll show you how to put buttons on the Contents Topic. Also, since we've already covered how to add a picture to the Contents Topic (see above), in this section we'll focus on replacing the simple underlined jump text with pushbuttons.

Looking Under the Hood

Placing pushbuttons on the Contents Topic will require directly editing the RTF file, which means you'll have to "look under the hood" to understand what's going on. Now we're not talking about changing the engine block or anything complicated; this will be more like checking the oil.

The RTF file that produced the simple help screen is shown below. The fact that the word "Introduction" is formatted with a double underline means that this text will appear in the help file as green underlined text, and that clicking on this text will jump to the topic specified by the hidden text code "Introduction.2." What we want to do is replace the word "Introduction" with a graphic button. The graphic button still needs to be formatted with a double underline for this to work and this button — like the text it is replacing — needs to be followed immediately by the context string of the topic to which you want to jump.



#·$·K·{bmc·help001.wmf}·Contents¶

OverviewOverview.2¶
Your·Pet's·DatabasesYourPet.sDatabases.3¶
Your·Pet's·PersonalityYourPet.sPersonality.4¶
Your·Pet's·EnvironmentYourPet.sEnvironment.5¶

#.$.K.Overview¶

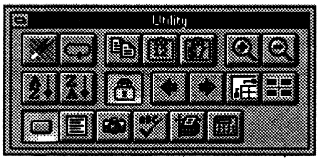
IntroductionIntroduction.6¶
Uses·For·the·ElectroPetUsesFortheElectroPet.8¶

#.$.K.Your·Pet's·Databases¶

Naming·Your·New·FriendNamingYourNewFriend.16¶
Friends·DatabaseFriendsDatabase.18¶

Click the Show All icon to reveal the hidden text.

Drawing the Buttons

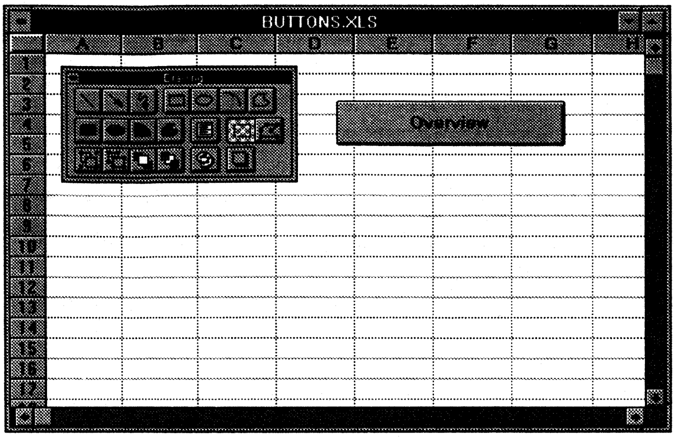


While practically any application with drawing tools will do, for this example we'll use Microsoft Excel to create the buttons because, well, Excel is good at making buttons. If you don't have Excel, you can use Paintbrush to create a simple rectangle with some text inside it.

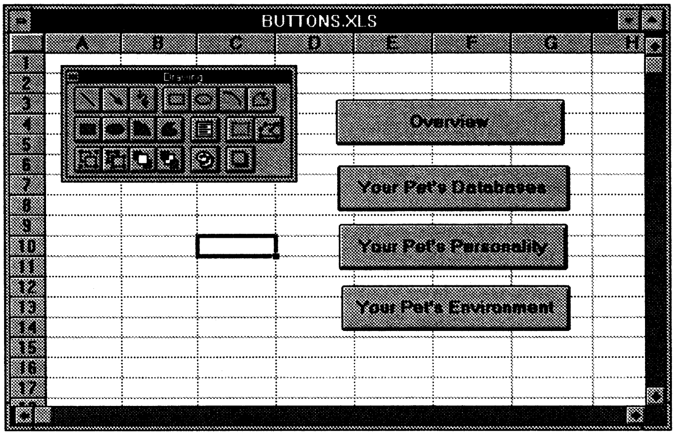
To create the buttons in Excel, we'll use the Macro Button tool. In Excel 3.0, this tool is on the standard toolbar. In Excel 4.0, the tool can be found on the Utility toolbar.

To Draw the Buttons

1. Draw a rectangle using the Macro Button tool. Excel will display the Assign Macro to Object dialog box.
2. Since we're not trying to work with Excel macros here, click on **Cancel**. At this point you'll have a nice looking button with the text "Button 1" in the button center.
3. Highlight this text and replace it with the word "Overview." Your screen should resemble the one shown below.



4. Using the Edit Copy command, copy and paste this button onto three different parts of the worksheet and enter the appropriate text, as shown below.



Copying the Buttons into Word

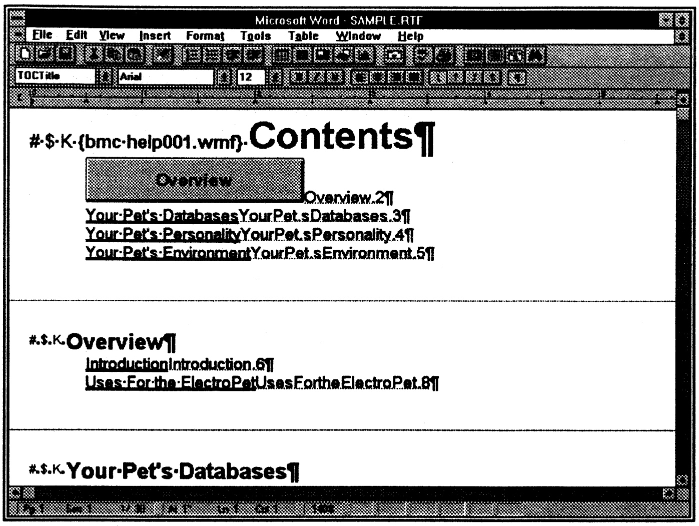
You're now ready to replace the jump text in the RTF file with buttons. You may want to save your Excel spreadsheet first because once you have copied the buttons into Word, you won't be able to copy them back into Excel to change the text on the buttons.

To Copy the Buttons from Excel

1. Select the first button you want to copy, in this case the "Overview" button.
2. From the **Edit** menu, choose **Copy**.
3. Activate **Word for Windows**.
4. Highlight the text you want to replace, in this case the double-underlined word "Overview."

Even though you can't see the double underline, if you select the button, click on Format and choose Character, you'll see that the button is still formatted with a double underline.

5. From the **Edit** menu, choose **Paste**. Your screen should look like the one shown below.

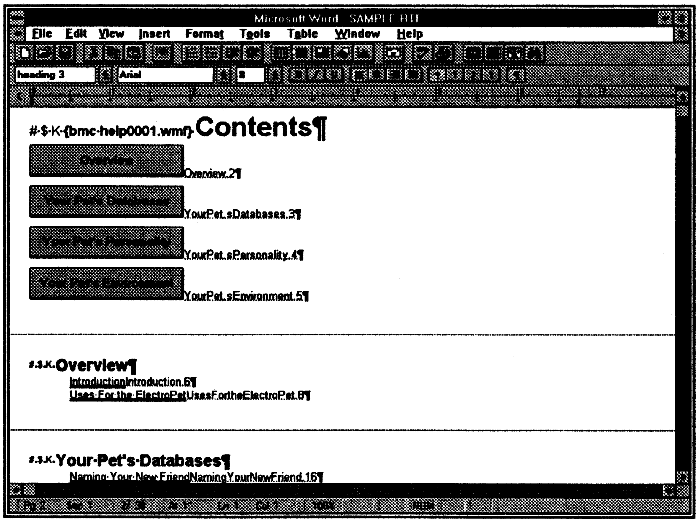

Microsoft Word - SAMPLE.RTF
File Edit View Insert Format Tools Table Window Help
TOCTitle
Arial
12
Overview

#·$·K·{bmc·help001.wmf}·Contents¶
Overview.2¶
Your·Pet's·DatabasesYourPet.sDatabases.3¶
Your·Pet's·PersonalityYourPet.sPersonality.4¶
Your·Pet's·EnvironmentYourPet.sEnvironment.5¶

#.$.K.Overview¶
IntroductionIntroduction.6¶
Uses·For·the·ElectroPetUsesFortheElectroPet.8¶

#.$.K.Your·Pet's·Databases¶

6. Repeat steps 1-5 for the other buttons. When you're finished, your screen should look like the one shown below.



#·$·K·{bmc·help0001.wmf}·Contents¶
Overview.2¶
YourPet.sDatabases.3¶
YourPet.sPersonality.4¶
YourPet.sEnvironment.5¶

#.$.K.Overview¶
IntroductionIntroduction.6¶
Uses·For·the·ElectroPetUsesFortheElectroPet.8¶

#.$.K.Your·Pet's·Databases¶
Naming·Your·New·FriendNamingYourNewFriend.16¶

Changing Spacing and Recompiling

We're almost ready to save the edited RTF file and recompile. However, before we do we'll probably want to add more space between the buttons. If we leave the spacing as it is, the buttons will be right on top of each other.

To Adjust the Spacing

1. Highlight the four buttons.
2. From the Format menu, choose Paragraph.
3. When the dialog box appears, change the spacing before to 3pt and the spacing after to 3 pt.
4. Close the Format Paragraph dialog box.

The only thing left is to save your work and recompile. To recompile, just choose Compile from the Tools menu. When the Compile dialog box appears, turn off the Write Help Project File option as the .HPJ file from the previous compilation will still be valid. (If you deleted the .HPJ file then you'll need Doc-To-Help to write a new one, so leave this option on.)

Some Words of Caution

You can use WexTech's Quicture to offload graphics to disk and replace them with {bmx} references. For information on Quicture, contact WexTech Systems.

Converting your original source document into help again will overwrite any changes you may make to the RTF file. If you plan to convert the source file into help again, copy your customized context topic from the RTF file into a separate file so that you can easily incorporate the changes into the newly converted RTF file. Also, if your pushbuttons consist of complicated graphics (that is, graphics that when saved are larger than 64K) you'll need to save your graphics to disk and refer to the graphics using a {bmc} reference field.

If you will be converting your source document into help again, you can easily automate the process of replacing the standard contents topic with the custom contents topic. For more information, see "User-Defined WordBasic Macros" on page 131.

Creating Your Own Help Topic

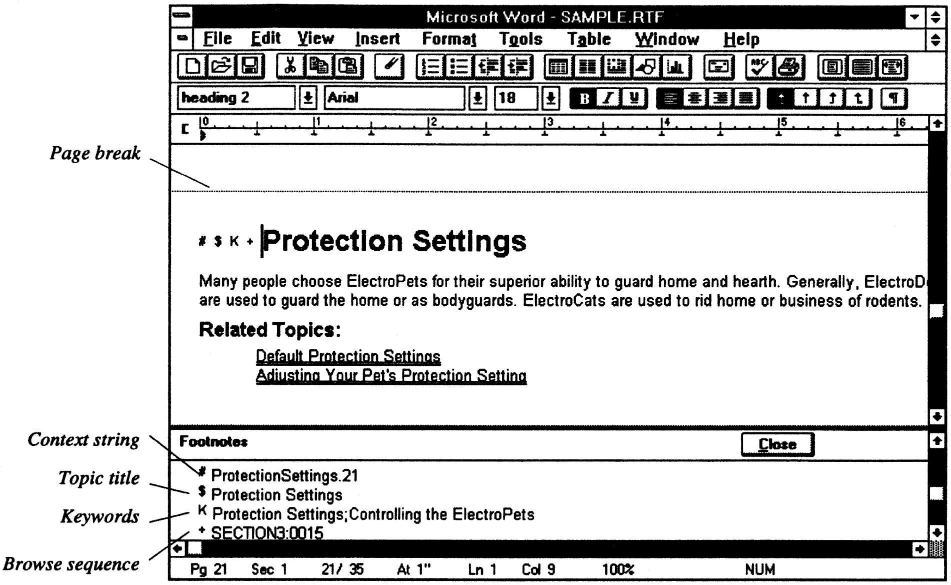
Perhaps the best feature of Doc-To-Help is that you produce your manual and Help file from the same source document. That is, if you change the manual, that change will appear in the Help file when you convert your manual into Help.

However, there may be times when you need to make a last minute change to the RTF file—either to insert a new link (which you can do using the Insert Related Help topic command) or to insert a whole new help topic. We use the term "last minute" because anything you add to the RTF file will not appear in the manual, so if you change the manual and then convert the manual into Help again, the edits you made to the RTF file will be lost.

If, however, you know you won't need to change the manual, or if you are in a hurry and don't want to wait for Doc-To-Help to convert the manual all over again, you can add a new topic to the RTF file.

Understanding the Footnote Codes

Throughout this manual we have attempted to shield the reader from the complexities of the RTF file. However, to be able to enter your help topics from scratch, you must "look under the hood" and understand the four basic footnote codes. Consider the following example:


Microsoft Word - SAMPLE.RTF
File Edit View Insert Format Tools Table Window Help
heading 2
Arial
18
Page break
Context string
Topic title
Keywords
Browse sequence
Footnotes
Close
Pg 21 Sec 1 21/ 35 At 1" Ln 1 Col 9 100% NUM

$ K + Protection Settings

Many people choose ElectroPets for their superior ability to guard home and hearth. Generally, ElectroD are used to guard the home or as bodyguards. ElectroCats are used to rid home or business of rodents.

Related Topics:

Default Protection Settings
Adjusting Your Pet's Protection Setting

ProtectionSettings.21
$ Protection Settings
K Protection Settings;Controlling the ElectroPets
+ SECTION3:0015

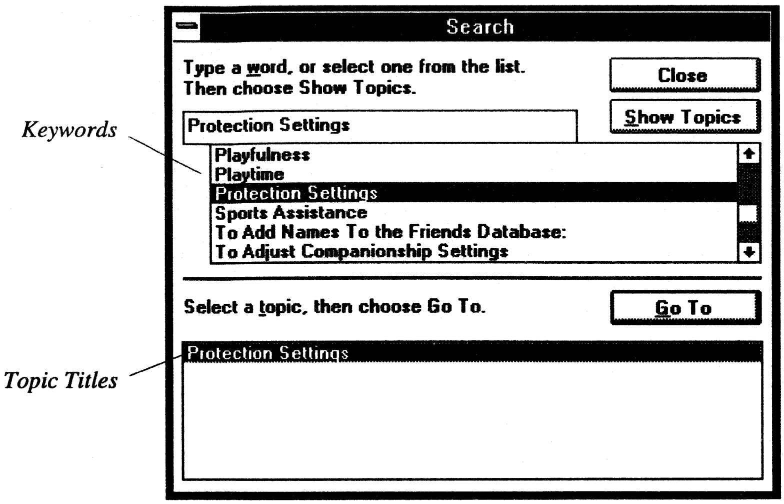
The screen shot above shows a section of the RTF file created from the tutorial. You will notice that the help topic "Protection Settings" is preceded by a hard page break and four footnote codes.

The Context String

The **Context String** (in this case "ProtectionSettings.21") is identified using a number sign (#) footnote. The context string is the most important of the footnote identifiers because it assigns a unique "address" to your help topic. It is through this unique address that the Windows Help engine can determine how to jump to this topic.

The Topic Title

The **Topic Title** (in this example "Protection Settings") is identified with a dollar sign ($) footnote. Windows Help displays this topic title whenever it refers to this topic. For example, Windows Help displays topic titles in the Topics Found list in the Search dialog box, as shown below.



Type a word, or select one from the list.
Then choose Show Topics.

Protection Settings

Select a topic, then choose Go To.

Protection Settings

Keywords

Keywords (in this example "Protection Settings" and "Controlling the ElectroPets") are identified using the uppercase K footnote. Keywords appear in the top portion of the search dialog box and provide a fast way to access a topic.

Browse Sequences

In examining the headings contained in your document, Doc-To-Help groups related subjects and displays these subjects in a specific order called a **Browse Sequence**. A browse sequence is a series of topics that appear in sequence when the user clicks the << and >> buttons.

Browse sequences are identified with a plus sign (+) footnote. In the example shown on page 122, "Protection Settings" is browse item 0015 in the SECTION3 browse sequence. By default, Doc-To-Help increments each browse item by 5, so the previous topic in this browse sequence is SECTION3:0010 and the next topic is SECTION3:0020. This way, if you want a topic to immediately follow "Protection Settings" in the browse sequence, you could give the new topic a sequence identifier between 0015 and 0020, such as SECTION3:0018.

Creating a New Topic

For this next example, let's suppose you want to create a new topic called "Maintenance." While you can place this new topic anywhere in the RTF file as long as a hard page break appears before the topic, for this example we'll place the topic at the end of the file.

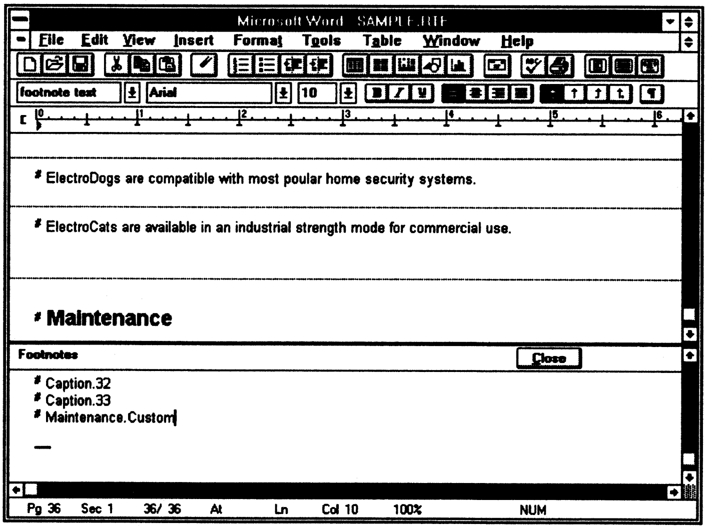
To Create a New Topic

1. Go to the end of the RTF file by pressing CTRL+END.
2. Insert a hard page break by pressing CTRL+ENTER.

3. Type the heading you want to appear, in this case "Maintenance."
4. Format the heading in either Heading 2, Heading 3 or Heading 4 style.
5. Press the **ENTER** key.
6. Type descriptive text below the heading.

To Insert the Context String

1. Place the cursor immediately to the left of the topic heading.
2. From the **Insert** menu, choose **Footnote**.
3. When the dialog box appears, enter a number sign (#) as the custom footnote and click **OK**. A superscript number sign will appear next to the heading and the cursor will move to the footnote window.
4. Type a unique context string next to the number sign footnote. For this example, type **Maintenance.Custom**. Your screen should look like the one shown below.



footnote text | Arial | 10

ElectroDogs are compatible with most poular home security systems.

ElectroCats are available in an industrial strength mode for commercial use.

Maintenance

Caption.32
Caption.33
Maintenance.Custom

Pg 36 Sec 1 36/ 36 At Ln Col 10 100% NUM

New Topic with Context String Footnote

Make sure only a single space exists between the # sign and the text string. The context string itself should not contain any spaces.

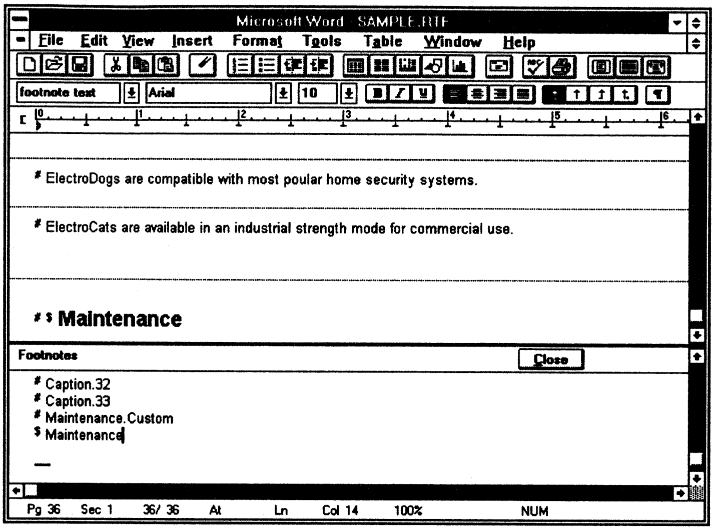
5. Click the **Close** button.

To Enter a Topic Title

1. Place the cursor immediately to the left of the topic heading.
2. From the **Insert** menu, choose **Footnote**.

3. When the dialog box appears, enter a dollar sign (**$**) as the custom footnote and click **OK**. A superscript dollar sign will appear next to the heading and the cursor will move to the footnote window.

4. Type the topic title as you want it to appear in the Search dialog box. For this example, type **Maintenance**. Your screen should look like the one shown below.



ElectroDogs are compatible with most poular home security systems.

ElectroCats are available in an industrial strength mode for commercial use.

$ Maintenance

Caption.32
Caption.33
Maintenance.Custom
$ Maintenance

Pg 36 Sec 1 36/ 36 At Ln Col 14 100% NUM

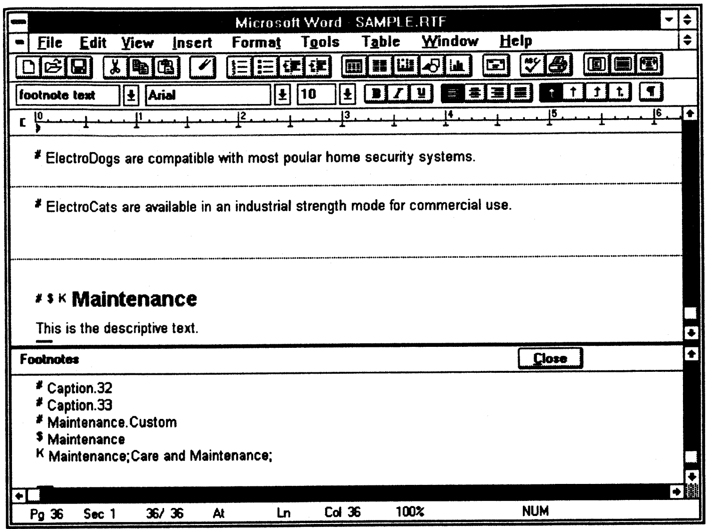
New Topic with Context String and Topic Title Footnotes

To Enter Keywords

1. Place the cursor immediately to the left of the topic heading.
2. From the **Insert** menu, choose **Footnote**.
3. When the dialog box appears, enter an uppercase **K** as the custom footnote and click **OK**. A superscript K will appear next to the heading and the cursor will move to the footnote window.
4. Type a keyword or phrase. If you have more than one keyword or phrase, separate them with semicolons (;) and place a semicolon after the last keyword. For this example, type

   Maintenance; Care and Maintenance;

   Your screen should look like the one shown below.


Microsoft Word - SAMPLE.RTF
File Edit View Insert Format Tools Table Window Help
footnote text
Arial
10
Footnotes
Close
Pg 36 Sec 1 36/ 36 At Ln Col 36 100% NUM

ElectroDogs are compatible with most poular home security systems.

ElectroCats are available in an industrial strength mode for commercial use.

$ K **Maintenance**

This is the descriptive text.

Caption.32
Caption.33
Maintenance.Custom
$ Maintenance
K Maintenance;Care and Maintenance;

New Topic with Context String, Topic Title and Keyword Footnotes

To Insert a Browse Sequence

1. Place the cursor immediately to the left of the topic heading.
2. From the **Insert** menu, choose **Footnote**.
3. When the dialog box appears, enter a plus sign (+) as the custom footnote and click **OK**. A superscript + will appear next to the heading and the cursor will move to the footnote window.
4. Type the browse sequence identifier, in this case SECTION3:0018.
5. Click the **Close** button.

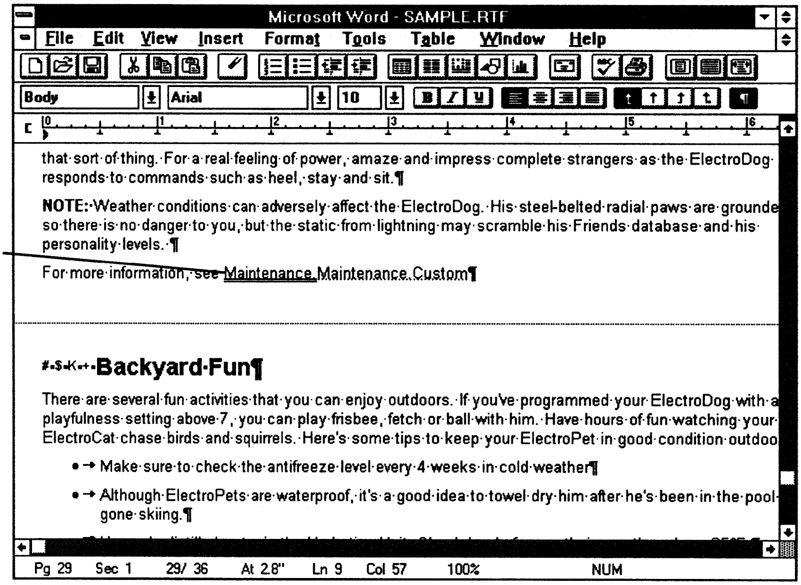
Linking to a Topic—Creating a Jump

In the previous sequence of steps, we saw how to create a new help topic by directly editing the RTF file. A topic inserted in this fashion can be accessed using the Search dialog box.

In addition to this search capability, you'll probably want to make your topic accessible by clicking on jump text, that is, green underlined text. To create a jump, the text you will want to make into a jump must be formatted with a double underline and must be followed by the context string of the topic to which you want to jump. This context string identifier must be formatted as hidden text.

To Create a Jump

1. Find the topic where you want the jump text to appear and type the jump text. In this example, type **For more information, see Maintenance.**
2. Using the **Format Character** command, apply a double underline to the text you want to make into a jump, in this case the word **Maintenance**.
3. Immediately to the right of the double underlined text, type the context string identifier of the topic you want to jump to, in this case **Maintenance.Custom**. Make sure that the context string is not formatted with a double underline.
4. Using the Format Character command, Format the context string identifier as hidden.
5. To see the hidden text, press CTRL+SHIFT+*. Your screen should look like the one shown below.


Microsoft Word - SAMPLE.RTF
File Edit View Insert Format Tools Table Window Help
Body
Arial
10
Pg 29 Sec 1 29/ 36 At 2.8" Ln 9 Col 57 100% NUM

that·sort·of·thing.·For·a·real·feeling·of·power,·amaze·and·impress·complete·strangers·as·the·ElectroDog· responds·to·commands·such·as·heel,·stay·and·sit.¶

NOTE:·Weather·conditions·can·adversely·affect·the·ElectroDog.·His·steel-belted·radial·paws·are·grounde so·there·is·no·danger·to·you,·but·the·static·from·lightning·may·scramble·his·Friends·database·and·his· personality·levels.·¶

For·more·information,·see·Maintenance.Maintenance.Custom¶

#·$·K·+·**Backyard·Fun¶**

There·are·several·fun·activities·that·you·can·enjoy·outdoors.·If·you've·programmed·your·ElectroDog·with·a playfulness·setting·above·7,·you·can·play·frisbee,·fetch·or·ball·with·him.·Have·hours·of·fun·watching·your· ElectroCat·chase·birds·and·squirrels.·Here's·some·tips·to·keep·your·ElectroPet·in·good·condition·outdoo

- → Make·sure·to·check·the·antifreeze·level·every·4·weeks·in·cold·weather¶
- → Although·ElectroPets·are·waterproof,·it's·a·good·idea·to·towel·dry·him·after·he's·been·in·the·pool· gone·skiing.¶

A jump is formatted with a double underline followed by the context string formatted as hidden.

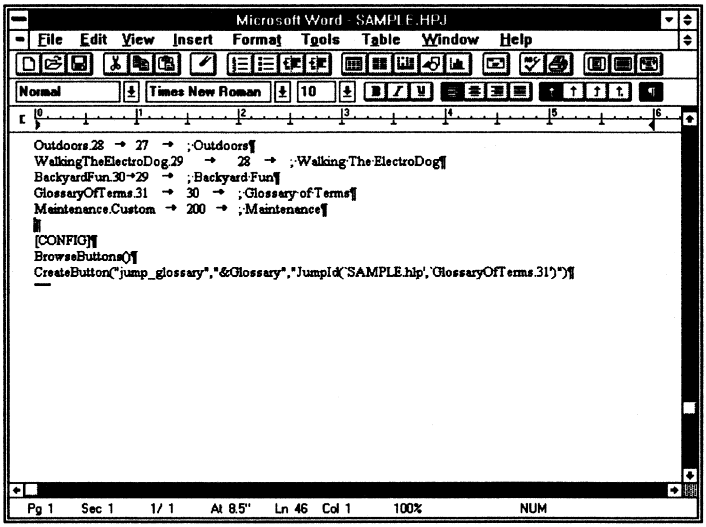
Making the Help Topic Accessible from Context-Sensitive Help

To make the newly-created help topic accessible from context-sensitive help, you will need to add a line to the [MAP] section of the HPJ file.

To Edit the HPJ File

1. Open the HPJ file found in your project directory.
2. Go to the bottom of the [MAP] section (probably located just before the [CONFIG] section).

3. Insert a new line.
4. Type the new topic's context string, in this case **Maintenance.Custom**, and press the TAB key.
5. Type a mapping number. Any number not already used will work, although the person who actually programs the context-sensitive help into your application may want a say in this matter.
6. Press the TAB key and type a semicolon (;) followed by a space and the topic title. The topic title following the semicolon is ignored by the Help compiler but is used by Doc-To-Help in creating the context mapping table. Your screen should look like the one shown below.


Microsoft Word - SAMPLE.HPJ
File Edit View Insert Format Tools Table Window Help
Normal
Times New Roman
10
Pg 1 Sec 1 1/ 1 At 8.5" Ln 46 Col 1 100% NUM

```
Outdoors.28 → 27 → ;·Outdoors¶
WalkingTheElectroDog.29 → 28 → ;·Walking·The·ElectroDog¶
BackyardFun.30→29 → ;·Backyard·Fun¶
GlossaryOfTerms.31 → 30 → ;·Glossary·of·Terms¶
Maintenance.Custom → 200 → ;·Maintenance¶
¶
[CONFIG]¶
BrowseButtons()¶
CreateButton("jump_glossary","&Glossary","JumpId(`SAMPLE.hlp',`GlossaryOfTerms.31')")¶
```

7. Save and close the HPJ file.

Formatting Source Code

Overview

If the document you write concerns programming or programming techniques, you will probably want to include source code samples. The Doc-To-Help templates contain three styles that will help you document source code: **Source**, **SourceTop** and **CodeExplained**.

Source and SourceTop

SourceTop is the same as Source, but with a little extra space added before.

The next example contains several lines of WordBasic source code. All lines are formatted using **Source** style except the top line, which is formatted using **SourceTop** style.

```
Sub MAIN
If SelType() = 1 Then
     EditGlossary .Name ="CaptionBody", .Context = 0, .Insert
     LineUp
FormatParagraph .Style ="BodyTable"
Goto Bye
End If

EditCut
EditGlossary .Name ="CaptionBody", .Context = 0, .Insert
LineUp 1
EditPaste
EditGoTo"\Cell"
FormatParagraph .Style ="BodyTable"

Bye:
End Sub
```

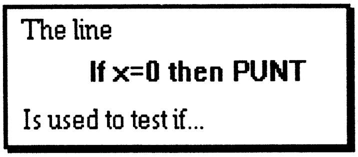
WordBasic source code in SourceTop and Source styles

Redefining Source Style

By default, Source style uses the same font as Heading 1 style, but without boldface. While this will probably suit your needs, there may be some occasions when using a proportional space font could cause confusion (hey, is that one space or two spaces?). If this happens in your document, you may want to redefine Source and SourceTop styles to employ a monospaced font such as Courier or Lineprinter.

CodeExplained

If your manual includes source code examples, the chances are very good that you may want to explain this code, one line at a time. Consider the example shown below:



The line

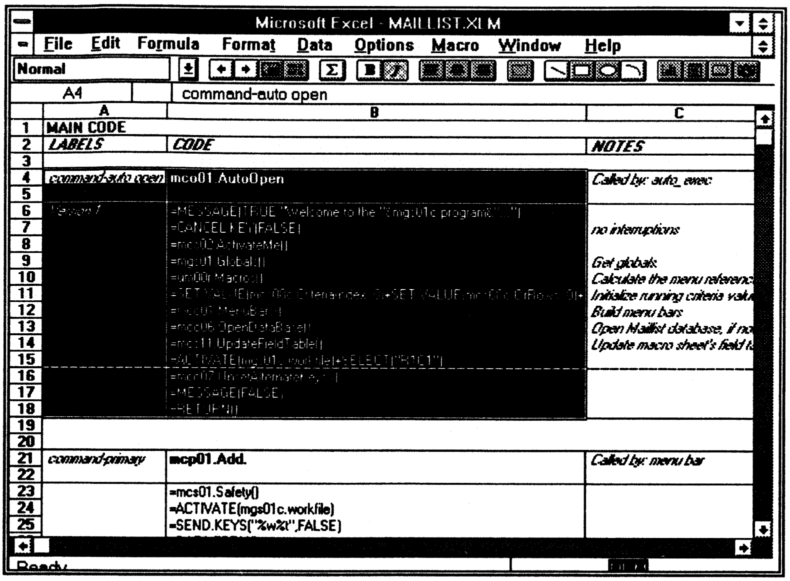
If x=0 then PUNT

Is used to test if...

Here the source code is formatted in CodeExplained style and the explanation is formatted in Body style.

Formatting Excel Source Code

You can produce particularly clear and readable Microsoft Excel macro code using Microsoft Excel's Copy Picture command. Let's suppose you want to incorporate the macro code shown below in a Doc-To-Help document.


Microsoft Excel - MAILLIST.XLM
File Edit Formula Format Data Options Macro Window Help
Normal
Ready

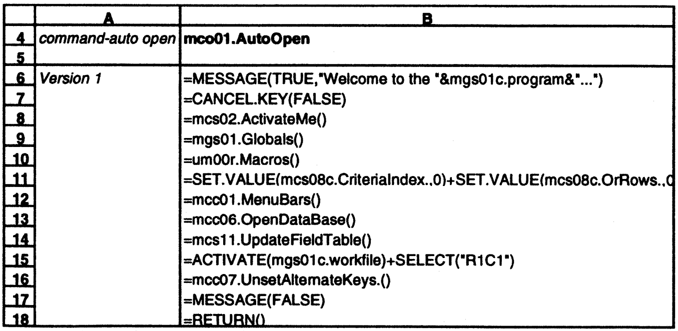
A4 | command-auto open

	A	B	C
1	MAIN CODE		
2	*LABELS*	*CODE*	*NOTES*
3			
4	*command-auto open*	mco01.AutoOpen	*Called by: auto_exec*
5			
6	*Version 1*	=MESSAGE(TRUE,"Welcome to the "&mgs01c.program&"...")	
7		=CANCEL.KEY(FALSE)	*no interruptions*
8		=mcs02.ActivateMe()	
9		=mgs01.Globals()	*Get globals*
10		=um00r.Macros()	*Calculate the menu referenc*
11		=SET.VALUE(mcs08c.Criteriaindex..0)+SET.VALUE(mcs08c.OrRows..0)+	*Initialize running criteria valu*
12		=mcc01.MenuBars()	*Build menu bars*
13		=mcc06.OpenDataBase()	*Open Maillist database, if no*
14		=mcs11.UpdateFieldTable()	*Update macro sheet's field t*
15		=ACTIVATE(mgs01c.workfile)+SELECT("R1C1")	
16		=mcc07.UnsetAlternateKeys.()	
17		=MESSAGE(FALSE)	
18		=RETURN()	
19			
20			
21	*command-primary*	**mcp01.Add.**	*Called by: menu bar*
22			
23		=mcs01.Safety()	
24		=ACTIVATE(mgs01c.workfile)	
25		=SEND.KEYS("%w%t",FALSE)	

To Insert Microsoft Excel Macro Code into a Word Document

1. Switch to Microsoft Excel and select the area to copy.
2. Make certain that **Row & Column Headings** is selected in the **File Page Setup** dialog box.
3. While holding down the **SHIFT** key, select **Copy Picture** from the **Edit** menu.
4. When the dialog box appears, select **As shown when printed** and click **OK**.
5. Switch to Word.
6. From the **Edit** menu, choose **Paste**. Word will insert an RTF picture of the Microsoft Excel source code, like that shown below. You may need to change the size using Word's Format Picture command.

Don't worry if the code appears unreadable on the screen. It will print fine.



	A	B
4	*command-auto open*	**mco01.AutoOpen**
5		
6	*Version 1*	=MESSAGE(TRUE,"Welcome to the "&mgs01c.program&"...")
7		=CANCEL.KEY(FALSE)
8		=mcs02.ActivateMe()
9		=mgs01.Globals()
10		=um00r.Macros()
11		=SET.VALUE(mcs08c.Criterialndex..0)+SET.VALUE(mcs08c.OrRows..(
12		=mcc01.MenuBars()
13		=mcc06.OpenDataBase()
14		=mcs11.UpdateFieldTable()
15		=ACTIVATE(mgs01c.workfile)+SELECT("R1C1")
16		=mcc07.UnsetAlternateKeys.()
17		=MESSAGE(FALSE)
18		=RETURN()

Microsoft Excel Macro Code

Copying Manuals

Doc-To-Help expects each manual to be kept in a separate directory. This directory is called the **Project Directory**. Each project directory contains the following files that make up your project:

- Main Document
- In a multifile project, any inside chapter files
- In a multifile project, the REMOVED directory
- DOC2HELP.INI
- D2H_COMP.BAT (if it exists)
- DOC2HELP.INF (if it exists)
- D2H_HEAD.BIN (if it exists)
- D2H_EXCL.TXT (if it exists)
- Files created when the document is made into Help: HPJ, ERR, HLP, RTF and the BITMAP subdirectory

If you want to move your manual to another directory, make sure that you copy *all* the files to the accompanying directory.

User-Defined WordBasic Macros

You should be comfortable with WordBasic before writing enhancements to the Doc-To-Help Make-Into-Help process.

By writing your own macros, you can augment Doc-To-Help's six-step "Make-Into-Help" process to include your own enhancements. These enhancements might include a process that automatically hard-codes certain jumps, a routine that temporarily hides certain graphics so they are not included in jump text, or a macro that changes the indent of tables.

The D2H_HELP template includes unencrypted skeleton macros for six user-defined macros that will be run at various points within the Make-Into-Help process, as shown below:

The Macro with this Name	Will Be Run at this Point
PrePreliminary	Before Doc-To-Help Preliminary Formatting
PostPreliminary	After Preliminary Formatting
PreJumps	Before Creating Jumps, Popups and Macros
PostJumps	After Creating Jumps, Popups and Macros
PreFinal	Before Final Formatting
PostFinal	After Final Formatting, but before Compiling

How User-Defined Macros Are Called

Before and after each process listed above, Doc-To-Help will make a call to the Main routine within the user-defined macro, passing the routine the document number of the active document. The syntax for this call is

Call UserDefinedMacro.Main(DocNum)

where DocNum is the number of the current document, as specified in the DocumentPrefs section of DOC2HELP.INI, and UserDefinedMacro is one of the six macros listed in the table shown previously. This number is passed to the Main routine in the user-defined macro so that the macro knows which document number it's working with. If you are working on a multifile project, Doc-To-Help runs the user-defined macros on all the files in the project.

User-Defined Macros and Multifile Projects

If you are working on a multifile project, Doc-To-Help will run the user-defined macros on each file in the project sequentially. Doc-To-Help runs all the related macros on a file before opening another file. For example, First Doc-To-Help opens the master document in a multifile project and runs the PrePreliminary, Preliminary and PostPreliminary routines. After these three routines have been completed, Doc-To-Help will open the next document in the project and run the three routines, and so on until all the documents in the project have been processed.

WordBasic and Technical Support

WexTech Systems is not responsible for any of the macros you create and will not field technical support questions on WordBasic.

The Doc-To-Help API

If you will be writing your own WordBasic macros to augment Doc-To-Help's Make Into Help process, you may find it useful to call some of Doc-To-Help's built-in macro subroutine and functions. What follows is a brief description of the functions and subroutines available through the Doc-To-Help Application Programming Interface (API).

GeneralLibrary.fActivateMasterDoc

Calling this function will make your project's master document the active document. This function is only available from the Master and Inside Chapter templates.

This function returns a -1 on success, 0 otherwise.

GeneralLibrary.fActivateMasterRTF

This function activates the project's master RTF file. This function is only available from the Help template.

This function returns a -1 on success, 0 otherwise.

GeneralLibrary.fCurDir$

Calling this function returns the name of the current directory.

This function returns a string. We provide this function because WordBasic's Files$(".") doesn't always return something you can change to using the ChDir command.

GeneralLibrary.fDocDir$

Calling this function determines the directory where the active document is saved.

This function returns the directory name as a string.

GeneralLibrary.fDocNumToDocName$(aDocNum)

Calling this function determines the name of a document, given the number of the document, as listed in the [DocumentPrefs] section of the DOC2HELP.INI file. For more information, see "What's in DOC2HELP.INI" on page 136.

This function returns the document name as a string.

GeneralLibrary.fDocs

Calling this function determines the total number of files in a multifile document.

GeneralLibrary.fGetPrivateProfileString$(Appname$, KeyName$, INIFile$)

Calling this function returns the value of KeyName$ in the Appname$ section of the private .INI file INIFile$. The INIFile$ argument must be the full path to an existing file. To obtain the full path to the DOC2HELP.INI file, use the fINIExists function.

If KeyName$ or Appname$ does not exist, this function returns a null string.

Example:

```
GeneralLibrary.fGetPrivateProfileString$( "DocumentPrefs",
"Multifile", "DOC2HELP.INI")
```

will return a 0 if the project is a single file project, but will return the number of inside documents if the project is a multifile project.

GeneralLibrary.fGetWordHandle

Calling this function will determine Word for Windows' application window handle. Doc-To-Help assumes that Word is active when the function is called and that Word is not a child of another application.

GeneralLibrary.fHPJ$

The full path includes the path and the file name.

This function returns the full path to the HPJ file that is associated with the active document. If the HPJ file for the current document does not yet exist, this function returns the full path of the HPJ file that *will* be created when the HPJ file is written. However, if the active document hasn't been saved, or the DOC2HELP.INI file is missing, the fHPJ$ will return a null string.

Example:

Let's say the current document is a named SUPERAPP.DOC, and it's part of a single-file project that's saved in c:\project. Executing this macro:

```
Variable$=GeneralLibrary.fHPJ$
```

will return

c:\project\superapp.hpj

GeneralLibrary.fIniExists(aIniPath$)

Calling this function determines if a DOC2HELP.INI file exists for the current document. The string argument aIniPath$ is set to the full path of the DOC2HELP.INI file for the active document.

The full path includes the path and the file name.

This function returns a -1 if DOC2HELP.INI exists, 0 otherwise. However, in executing the function, aIniPath$ is set to the full path of where the DOC2HELP.INI file should be, even if the file does not exist. If the active document has not been saved, this function will change the value of aIniPath$ to be a null string.

GeneralLibrary.fMultiFile

Calling this function determines whether the current document is part of a multifile project.

This function returns -1 if the current document is part of a multifile document, otherwise 0.

GeneralLibrary.fSearchForWindow(Search$, MatchType, ActivateNow)

Calling this function determines if there is a document open with a window name that contains the search string *Search$*.

MatchType determines the search criteria, as described below:

MatchType	Finds This
1	First window containing the search string.
2	Exact match.
3	First window matching the search string starting from the right of the window name. This will find a document regardless of the path.
4	Exact match, but ignores multiple instances (e.g., FILE.DOC:2)

If ActivateNow is set to -1 when the function is called, the function will activate the window, if a window is found. If ActivateNow is set to any other value, the function will change the value of ActivateNow to the number of the window that is found. This allows you to activate the window later, if you want.

This function returns -1 if a window is found.

GeneralLibrary.fGlobalINIPath$

The full path includes the path and the file name.

This function determines the full path to the D2H_GLBL.INI file. D2H_GLBL.INI contains global Doc-To-Help settings such as the Word for Windows language version you are using, indexing preferences and the maximum number of open files.

This function returns a string.

GeneralLibrary.sWritePrivateProfileString(Appname$, Keyname$, Value$, INIFile$)

This subroutine sets a variable in a private .INI file. Specifically, this routine will set the *Keyname$* to *Value$* in the *Appname$* section of the file *INIFile$*. For example, calling

```
If GeneralLibrary.fINIExists(IniPath$) then
   Call GeneralLibrary.sWritePrivateProfileString(
"DocumentPrefs",  "Index",  "1", IniPath$)
End If
```

will make the following setting within your project's DOC2HELP.INI file:

```
[DocumentPrefs]
Index=1
```

GeneralLibrary.fExistingMacro(aName$, aContext)

This function determines whether a macro named *aName$* exists in a particular context (*aContext*). If aContext = 0, then the context is global; otherwise, the context is template.

What's in DOC2HELP.INI

The DOC2HELP.INI file is divided into several major sections. Depending on the project, it's possible that not all of the sections listed below will be found in a DOC2HELP.INI file.

[DocumentPrefs]

Keyname	Meaning
MultiFile=n	This is always one less than the total number of documents in a help project, including the master document. In a single-file project, MultiFile is always zero. In a help project that contains three chapters in addition to the master document, MultiFile will equal 3.

Doc1=MASTER.DOC
Doc2=MARKET.DOC
Doc3=WINNING.DOC
	Names of all documents in the help project are listed in the DocumentPrefs Section in this form.

Title=Functional Specifications for Doc-To-Help	This is a setting that is stored based on what you type in the Title field in the Set Project Options dialog.
BrowseInterval=15	This allows you to specify an interval between help topics in a browse sequence. The default is five.

[Search Item Defaults]

These are the defaults for the four corresponding checkboxes in the Index Target Options dialog box, which determine the default settings for a new index target. These settings are always equal to either zero or one.

Keyname	Meaning
MatchCase=0	This specifies whether the search for the index target is case sensitive.
WholeWord=1	This specifies whether the Whole Word Only parameter is used in the search.
PromptEachInstance=0	This specifies whether the index target is Auto (0) or Discretionary (1).
CreateEntriesAt=0	This specifies that after finding an index target, the next search takes place: (0) at the top of the following page, (1) at the beginning of the following chapter, (2) just after the last instance, or (3) at the beginning of the next heading paragraph.

[D2HStatus]

This maintains the status of the Make Into Help process. In most cases, a value of "1" indicates that the specified phase is complete.

KeyName	Meaning
ContentsHeadings=1	Specifies whether the Contents Screen is to contain Heading 1 paragraphs only (1), or Heading 1 and Heading 2 paragraphs (0).
ConvertBitmaps=1	Specifies whether or not the part of preliminary reformatting which offloads embedded graphics to disk has completed.
ApplyNewStyles=1	Specifies whether the macro which applies the styles from the Help template has completed.
RemoveReservedFormatting	Specifies whether the macro which removes reserved formatting (such as Manual-Only red text) has completed.
AutoRelatedTopics=1	Specifies whether the AutoRelatedTopics macro, which creates hypertext links and macros, has completed.
DoMainTopicLinks=1	Specifies whether the DoMainTopicLinks macro, which creates the contents topic, has run yet.
DoMainTopics=1	Specifies whether the macro DoMainTopics, which converts headings into help topics, has completed.
GlossaryDefinitions=1	Specifies whether the GlossaryDefinitions macro, which converts Glossary terms into popup definitions in the help file and creates the Glossary topic, has completed.
FinalReformatting=1	Specifies whether the FinalReformatting macro has completed.
CaptionsToDefinitions=1	Specifies whether the CaptionsToDefinitions macro, which converts captions into popups, has completed.
RemoveTableFormatting=1	Specifies whether the RemoveTableFormatting macro, which converts tables into plain text for the Windows 3.0 help compiler, has completed.

[D2HTitlePage]

These are the settings for the title page, which are retained in case the title page is removed or rendered unreadable by the Set Project Options dialog.

```
Title=Doc-To-Help
SuperTitle=Functional Specifications for
Byline=By WexTech Systems, Inc.
```

[Index Target]

Each Index Target has its own section in the DOC2HELP.INI file, where index target options and index tags associated with the index target are stored.

Keyname	Meaning
MatchCase=n	Corresponds to the Match Case search criterion specified in the Index Target Options dialog.
PromptEachInstance=n	Specifies whether the index target is Auto (0) or Discretionary (1).
WholeWord=n	Specifies whether the Whole Word Only search criterion is in effect for this target.
CreateEntriesAt=n	Corresponds to the "Place Index Tags At" option. This specifies that after finding an index target, the next search takes place: (0) at the top of the following page, (1) at the beginning of the following chapter, (2) just after the last instance, or (3) at the beginning of the next heading paragraph.
NumEntries=n	This is the total number of index tags associated with the Index Target, including tags which have been deleted.
NumDeleted=0	This is the number of index tags which have been deleted. Index tags marked for deletion are maintained as placeholders so they can be eliminated properly when the index is finalized.

Entry1=auxiliary¦1
Entry2=Frederick¦1
Entry3=##DELETED##
	Each Index Tag is listed in this fashion, followed by an ANSI 166 character and a one or a zero, where a one specifies that the tag is designated as a default tag for that index target, and a zero indicates a non-default tag. When an index tag is deleted, it is changed to ##DELETED## in the DOC2HELP.INI file. This is to account for temporary index codes which may already have been inserted into the document before the index was finalized.

[Headings]

Keyname	Meaning
NextBookmark=2	This specifies the number to use for the next bookmark created by the Insert Hypertext Link command or the Insert Help Macro command (InsertHelpMacro).

[Search List]

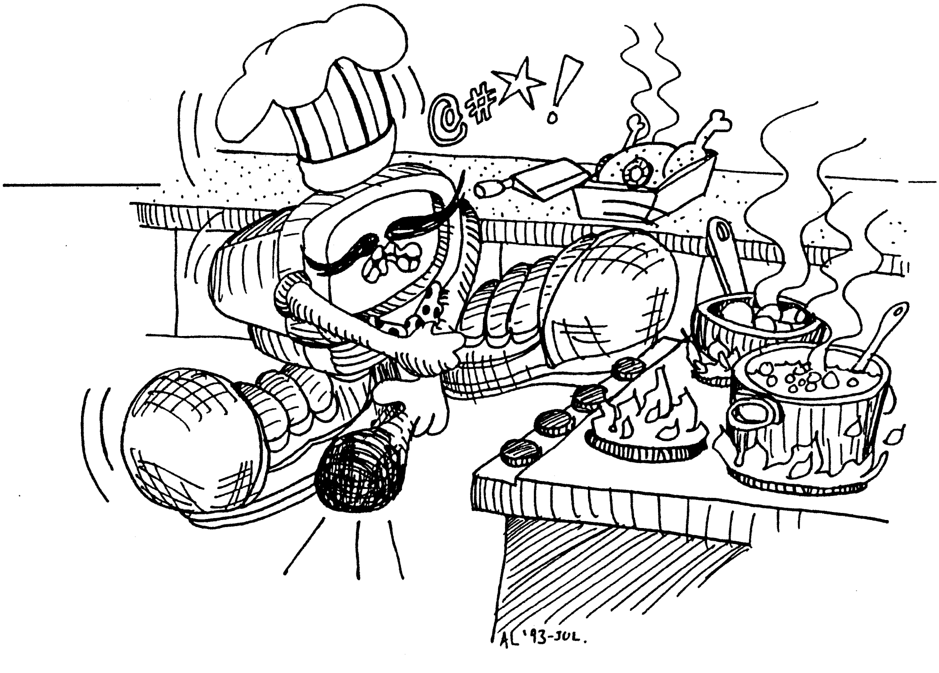
This stores information about the Index Target List.

Keyname	Meaning
NumSearchItems=n	This is the total number of Index Targets contained in the list.
SearchItem1=text text text	Each index target is listed in this form.
NewItemsExist=n	If n=1 then the Index Target List has to be sorted when the Edit Index Target List command (EditIndexList macro) is run.

[MakeIntoHelp]

Keyname	Meaning
MapHex=n	If n=1, Doc-To-Help will write map numbers in hexadecimal rather than decimal format.

Hypergraphics



Hypergraphics

Put Some Hot Spots in Your Life with SHED.EXE

Many Doc-To-Help users have called to find out if they could add multiple hot spots to their graphics. Some had seen help files with graphics that users could click on to jump to other screens or to see a popped-up explanation and wondered how that was done. Others knew the secret but wondered how they could use it with Doc-To-Help. The secret is a utility from Microsoft called SHED.EXE, the Segmented Hypergraphics Editor. Not only can you use SHED with Doc-To-Help, it is included with release 1.5.

What Is a Segmented Hypergraphic Anyway?

Think of how useful it would be to include a sample of a toolbar in your help file and have each button offer a popup description of its function, or a database screen or dialog box where each field or text box does the same. SHED.EXE, the Segmented Hypergraphics Editor, allows you to embed multiple segments, or hot spots, in a graphic that can be associated with jumps, popups or macros. The source graphic can be either a bitmap (BMP) or a Windows Metafile (WMF). SHED saves these modified bitmaps/metafiles in an .SHG (***s***egmented ***h***yper***g***raphic) format that can be compiled as part of your help file.

How Do I Get SHED to Work with Doc-To-Help?

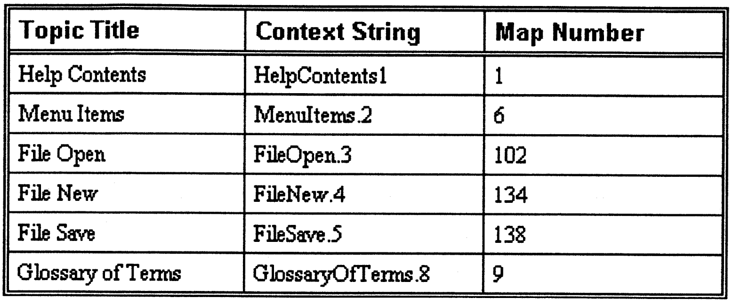
Overview

You will not find any menu option in the current version of Doc-To-Help for using the hot spot editor, but you can still use it, and it's easy.

First, position your graphic in the Doc-To-Help document as you would normally, using Edit Paste or Insert Picture.

Create the RTF file by choosing Format, Make into Help and click "Yes" in the dialog box that asks "Do you want to compile the file into help after reformatting?" When the Compile dialog box appears, de-select "Run the Help Compiler" in the Compile dialog box and choose "Write Help Project File" (HPJ). Click OK.

From your RTF file, choose Tools, Show Context Mapping, and print out the table that lists all your topics with their corresponding context strings and map numbers (this information is extracted from the [MAP] section of the HPJ). You will need this information for the SHED utility. A sample context-mapping table is shown below.



Topic Title	Context String	Map Number
Help Contents	HelpContents1	1
Menu Items	MenuItems.2	6
File Open	FileOpen.3	102
File New	FileNew.4	134
File Save	FileSave.5	138
Glossary of Terms	GlossaryOfTerms.8	9

Example of a table produced using the Show Context Mapping command

If you examine the RTF file, you'll notice that all graphics (including the one you want to create hot spots in) have been removed from your RTF file and referenced with a "bmc" reference. Doc-To-Help does this because the Microsoft Help Compiler can handle a maximum of 64 KB per help topic and many graphics are larger than this. The removed graphics have been named sequentially (HELP0001 and so forth, with BMP or WMF extensions) and placed in a BITMAPS sub-directory under your project directory.

Loading SHED

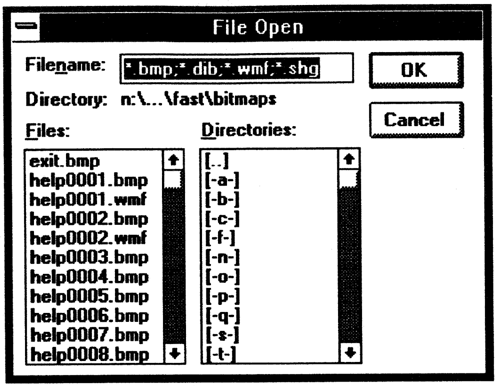
SHED.EXE and SHED.HLP, its associated help file, will be copied by the Doc-To-Help setup routine to your Windows directory unless you have specified another location. You can add SHED to your Program Manager by doing the following:

1. Activate the group window where you want to add the icon.
2. From the **File** menu, choose **New**.
3. Select **Program Item** in the **New Program Object** dialog box.
4. Type in **SHED.EXE** in the Command Line text box (or Browse for this file if you loaded it to a directory other than Windows.)
5. Click **OK**.

Using SHED

Now comes the fun. Double-click the SHED icon (or run SHED.EXE any other way you wish.) With SHED loaded, click **Open** from the **File** menu, and browse for the file you want to edit in the BITMAPS subdirectory.

You should see something similar to the screen shown below.



Filename: *.bmp;*.dib;*.wmf;*.shg

Directory: n:\...\fast\bitmaps

Files:

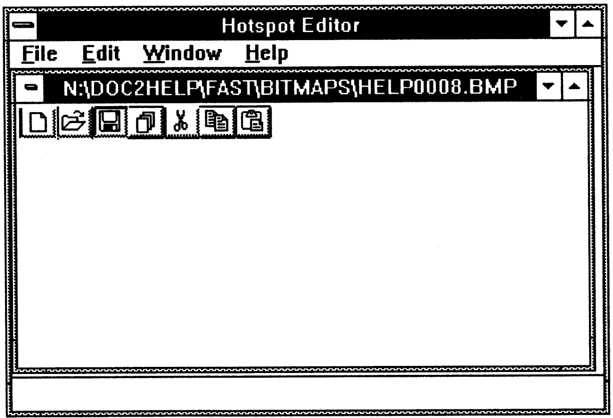
exit.bmp
help0001.bmp
help0001.wmf
help0002.bmp
help0002.wmf
help0003.bmp
help0004.bmp
help0005.bmp
help0006.bmp
help0007.bmp
help0008.bmp

Directories:

[..]
[-a-]
[-b-]
[-c-]
[-f-]
[-n-]
[-o-]
[-p-]
[-q-]
[-s-]
[-t-]

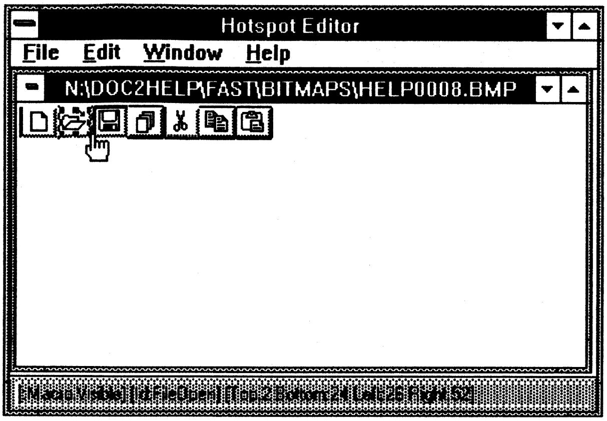
SHED File Open Dialog Box

In the example that follows, we've opened help0001.bmp, which contains a screen capture of part of the Word for Windows toolbar.



The Segmented Hypergraphics Editor with a simple bitmap loaded

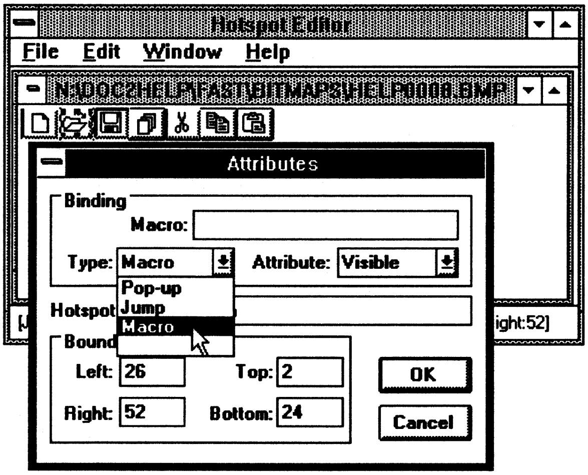
With the graphic you want to edit loaded, hold down the left mouse button and draw a rectangle around the segment you want to turn into a hot spot, as shown below.



[Macro Visible] [idFileOpen] [Top:2 Bottom:24 Left:26 Right:52]

Drawing a hot spot

Now double-click on the rectangular bounding box, and the SHED Attributes dialog box appears, as shown below.



Macro:

Type: Macro

Attribute: Visible

Hotspot

Bound

Left: 26 Top: 2

Right: 52 Bottom: 24

ight:52]

The Attributes Dialog Box

You are given the choice to bind (or link) the hot spot to either a Popup, Jump or Macro. If you choose Popup or Jump, you then will be asked to insert a Context String to associate with the hot spot. Context strings are what tie a help file together. One is created for each Topic, Heading 1 through Heading 4, each Glossary Term and each Caption. If you choose Macro, you can use any of the many help macros available (more about these later) including macros to produce jumps or popups. Users of Doc-To-Help Release 1.5 and earlier are advised to choose the Macro option here. If you must use the Jump or Popup options, be aware that the context strings provided will be lost upon re-converting the manual into an RTF file, and you'll have to specify them again. Luckily there are jump and popup macros that use the map number of a topic to create the link, and map numbers are kept constant in Doc-To-Help.

Inserting a JumpContext Macro (JC Macro)

In this example, we want to make a jump from the File Open button of the toolbar to a topic that describes that menu item (map number 102), which was among those we noted earlier in our Show Context Mapping document. The macro that we will use to achieve this is the JumpContext macro. Its syntax is as follows:

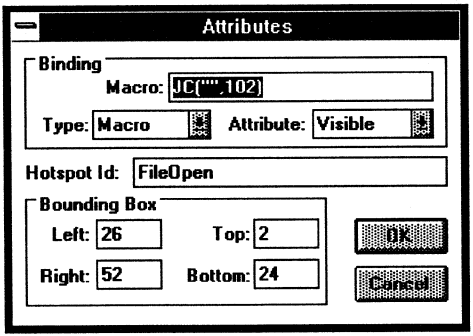
JumpContext("*filename.hlp*",mapnumber)

or even more simply:

JC("",mapnumber)

If the jump is to a topic in the same help file, you will note that you do not have to include a filename.

The screen below shows this macro inserted. Note that you may want to change the Attribute to Invisible so that a visible rectangle does not appear around the "hot" area in the help file.



Macro: JC("",102)

Type: Macro

Attribute: Visible

Hotspot Id: FileOpen

Left: 26

Top: 2

Right: 52

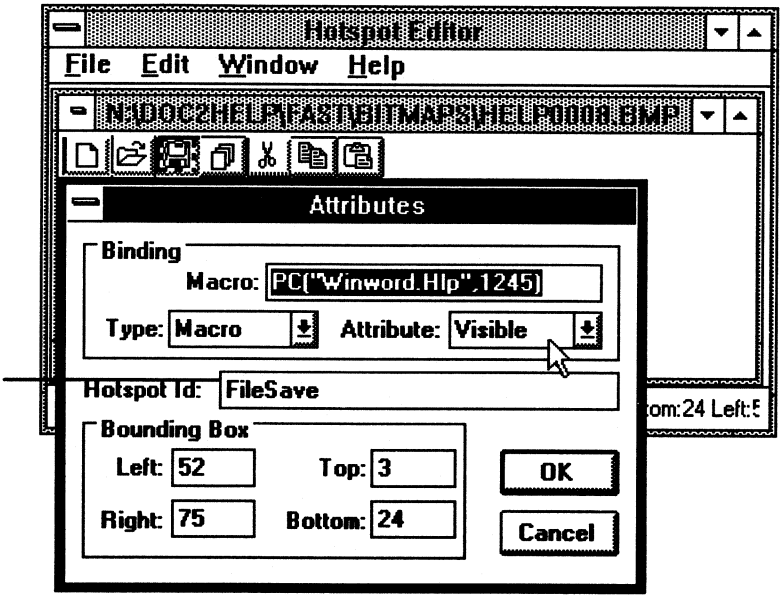
Bottom: 24

Attributes dialog box with JumpContext macro

We now could save our graphic, but let's add a popup. In this case, a popup to a topic in another help file.

Inserting a Popup Macro (PC Macro)

1. With the graphic you want to edit loaded into SHED, draw a rectangle around the area you want to make into a popup.
2. Double-click on the rectangular region to access the Attributes dialog box.
3. Change the Binding type to Macro and fill in the macro attributes. An example is shown below.



Macro: PC("Winword.Hlp",1245)
Type: Macro
Attribute: Visible
Hotspot Id: FileSave
Bounding Box
Left: 52
Top: 3
Right: 75
Bottom: 24

Each hot spot can be given a name

Inserting a Popup Macro

The syntax is as follows:

PopupContext("*filename.hlp*",mapnumber)

or

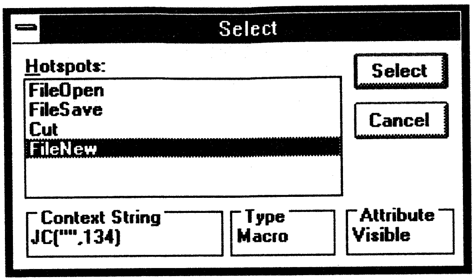
PC("*filename.hlp*",mapnumber)

Remember, if the topic was in the same help file, you just need empty quotes.

A name can be added for each hot spot in the Hot Spot Id box. This will be useful for keeping track of multiple hot spots in your file and when you need to edit a list of hot spots and their attributes.

Editing a List of Hot Spots

Choose Select from the Edit Menu to see a list of all your hot spots by name. By selecting each, you can review its links. By double-clicking on one, you can then edit any of its attributes.



Hotspots:
FileOpen
FileSave
Cut
FileNew

Context String: JC("",134)
Type: Macro
Attribute: Visible

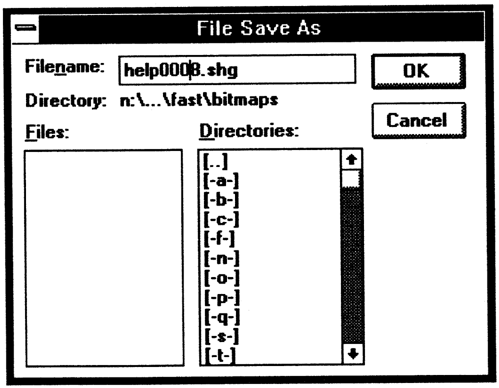
Select Dialog Box

Saving Your Work

After you have added all your hot spots and linked them appropriately, you'll be ready to save your work.

To Save a Bitmap as an SHG File

1. From the **File** menu, choose **Save As**. The Save as dialog box will appear, as shown below.



Filename: help0008.shg
Directory: n:\...\fast\bitmaps
Files:
Directories:
[..]
[-a-]
[-b-]
[-c-]
[-f-]
[-n-]
[-o-]
[-p-]
[-q-]
[-s-]
[-t-]

File Save As Dialog Box

2. Save your file to the BITMAPS subdirectory with the same name as the original BMP or WMF file, but with the default SHG extension.
3. Open the RTF file containing the original bmc reference for the BMP or WMF file you edited with SHED.
4. From the **Edit** menu, choose **Find** and search for the bmc reference, e.g., {bmc help0008.bmp}.
5. Change the extension to SHG, e.g., {bmc help0008.shg}.
6. Save your RTF file.

Compiling the Help File

Now that you have created the segmented hypergraphic and saved it to the BITMAPS directory, all you have to do is compile the HLP file from your RTF file. Choose Compile from the Tools Menu, select "Run Help Compiler," and Click OK. Open the topic that contains your hypergraphic and move your cursor over it. Each time you hit a hot spot, the arrow cursor will change to a pointing hand.

Placing SHG Files into the Manual

The method described above requires you to manually edit the bmc references in the RTF file. While making such changes is very simple, having to do so every time you make your manual into help can be somewhat tedious.

You may be wondering if there is a way to place SHG files directly into your manual. Regrettably, Microsoft does not supply a filter for SHG graphic files, but you can enter *references* to SHG files directly into your manual. Here's how you might go about doing this.

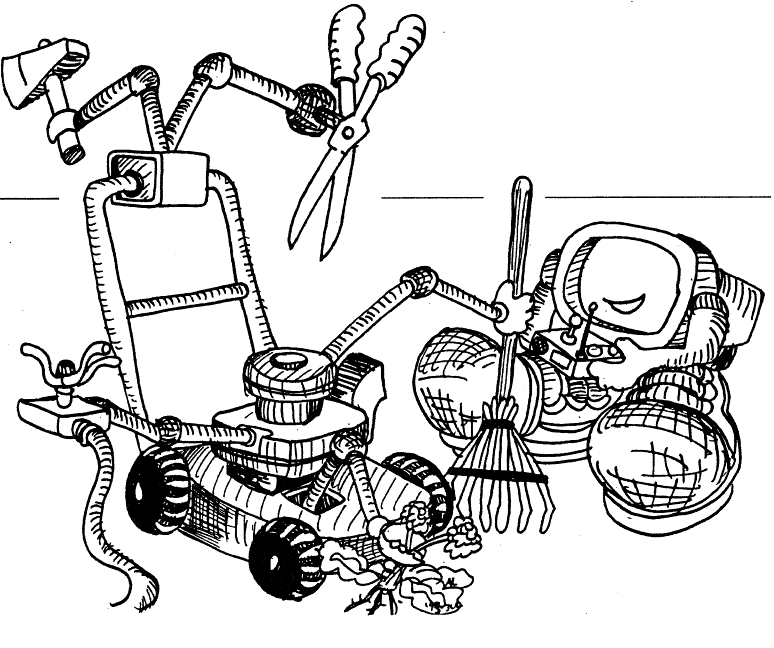
With Quicture the process for doing this is more straightforward as you only need to offload the graphics to which you will be adding hotspots; however, this option does require that you purchase an additional utility.

1. Use either WexTech Quicture or Doc-To-Help to offload your graphics to disk.
2. Edit the graphics to which you want to add hotspots using SHED, saving these graphics with an SHG extension in the BITMAPS directory below the current project directory.
3. Go back to the manual and find the first graphic to which you want to add hotspots.
4. In front of this graphic, enter a bmc reference to the SHG file that corresponds to the graphic in your manual. The syntax for the bmc reference is this:

   {bmc filename.shg}

   Make sure to use lowercase letters.
5. Mark this reference as For Help Only by selecting the text and pressing CTRL+SHIFT+H. (For more information, see "Format Doc-To-Help Markings" on page 226.)
6. Mark the graphic as For Manual Only by selecting the graphic and pressing CTRL+SHIFT+M.

Help Macros



Help Macros

Overview

Doc-To-Help provides easy access to any of the more than fifty macros which you can use to enhance the capabilities of your Help system. Macro uses include:

- Modifying the WINHELP button bar and menus.
- Adding keyboard shortcuts to WINHELP commands.
- Jumping to other Help files.
- Launching Windows applications.
- Registering a function in a Dynamic Link Library (DLL) and calling that function from a help macro. Through Dynamic Link Libraries you can extend the functionality of WINHELP to include playing sound and video files.

How Do Help Macros Get Run?

Basically, there are three different ways you can run a help macro.

Hotspot Macros

Hotspot macros look like jump text, but when you click on them they execute Help macros. Examples of Hotspot macros include text that when clicked runs another Windows application or prints the current help topic.

Startup Macros

A Startup Macro is run whenever the user loads the help file. Examples of Startup macros include adding new buttons and menus, registering a function or functions in a Dynamic Link Library (DLL) and playing sound. The settings for Startup macros are stored in the [Config] section of the Help Project File (HPJ File). If you examine an HPJ file, you'll see that Doc-To-Help uses Startup macros to add Browse buttons and a Glossary button to the Help button bar.

Help Topic Macros

You can create Windows Help macros that start whenever the user enters a certain topic. This is useful for customizing topics with buttons, menu items, and markers. For example, you may want certain buttons and menu items to be available only when a certain topic is displayed. Or perhaps you want to play a particular sound when displaying a topic.

Creating Hotspot Macros—Launching an Application

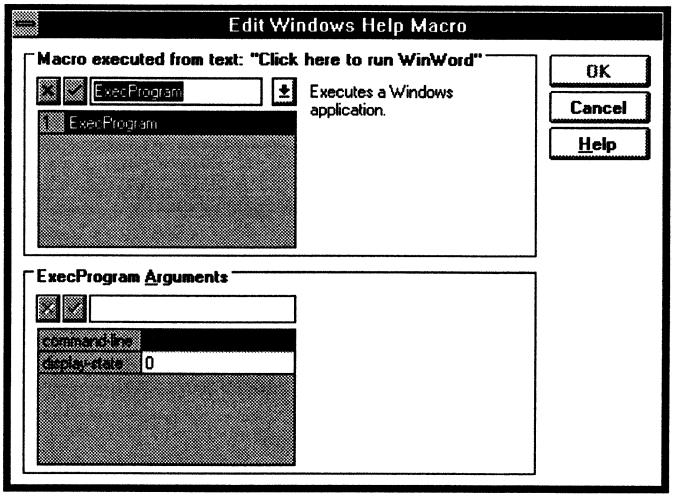
The following example makes text into a hotspot that when clicked will run Word for Windows.

To Create a Hotspot Macro that Launches Word for Windows

1. Highlight the text or graphic you want to make into a Hotspot macro.
2. From the **Insert** Menu, choose **Help Macro**. The Edit Windows Help Macro dialog box appears.

You can also use ExecProgram shortcut, EP.

3. The macro for running an application is called ExecProgram. From the drop-down list box, select **ExecProgram**. The Arguments group box will appear. Your screen should look like the one shown below.



Edit Windows Help Macro

Macro executed from text: "Click here to run WinWord"

ExecProgram

Executes a Windows application.

1 ExecProgram

ExecProgram Arguments

command-line

display-state 0

4. In the command-line box, type **winword.exe**.
   - If you don't want Word to open a new document, type **winword.exe /n**.

When specifying a path, enter two backslashes.

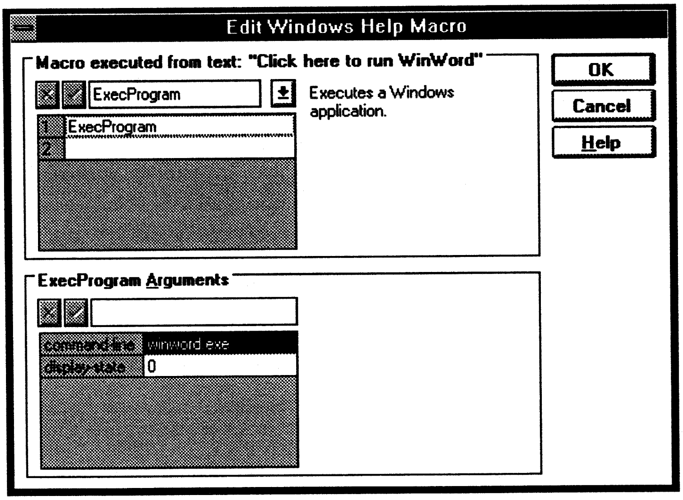
   - If you want to open a particular document in Word, leave a space after the winword.exe statement and type the document name (full path). For example, if you wanted to open a document named tutorial.doc located in the learn directory on drive C, you would type **winword.exe c:\\learn\\tutorial.doc**.
   - If you want to use another application, just use the executable (.exe) program name.

5. In the display-state box, select a value. Display-state determines how the application appears when it is started, as described in the table below.

Value	Action
0	Displays the application window sized as it would be if the user had launched the application. This is the default setting.
1	Minimizes the application and displays it as an icon.
2	Enlarges the application window to its maximum size.

6. Click the checkmark button in the ExecProgram Arguments section. This completes the arguments.

7. Click the checkmark button in the "Macro executed from" section. This completes the macro definition and inserts a new cell where you can enter another macro. Your screen should look like the one shown below.



Macro executed from text: "Click here to run WinWord"

ExecProgram

Executes a Windows application.

1 ExecProgram
2

ExecProgram Arguments

command-line	winword.exe
display-state	0

8. Click **OK** to place the macro field codes in your document. The macro exists as a field code until you make your document into Help. (For more information, see "Hypertext Links and Help Macros Don't Show Up in Help" on page 178.)

Creating Hotspot Macros—Printing a Topic

The following example makes text into a hotspot that when clicked will print the current help topic.

To Create a Hotspot Macro that Prints the Current Topic

1. Highlight the text or graphic you want to make into a Hotspot macro.
2. From the **Insert** Menu, choose **Help Macro**. The Edit Windows Help Macro dialog box appears.
3. The macro for printing a topic is Print. From the drop-down list box, select **Print**.
4. Click the checkmark button. This completes the macro definition.
5. Click **OK** to place the macro field codes in your document. The macro exists as a field code until you make your document into Help. (For more information, see "Hypertext Links and Help Macros Don't Show Up in Help" on page 178.)

Creating Hotspot Macros—Displaying a Popup from Another File

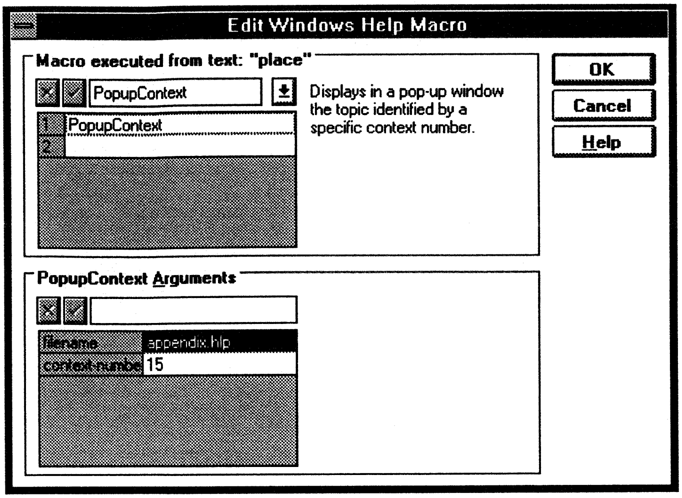
You can also use hotspot macros to perform certain help-specific commands. The following example displays a topic named "Tools," which is located in a file named appendix.hlp. The topic is displayed in a popup window when the user clicks the Hotspot macro text.

To Create a Hotspot Macro That Displays a Popup from Another Help File

1. Highlight the text or graphic you want to make into a Hotspot macro.
2. From the **Insert** Menu, choose **Help Macro**. The Edit Windows Help Macro dialog box appears.
3. The macro for displaying a popup in another file is PopupContext. From the drop-down list box, select **PopupContext**. The PopupContext arguments group box will appear.

You can also use the PopupContext shortcut, PC.

4. Type **appendix.hlp** in the filename argument box and click on the checkmark button.
5. Type a valid number in the context-number argument box and click on the checkmark button.
6. Click on the Macro executed from checkmark button. Your screen should look like the one shown below.



Macro executed from text: "place"

PopupContext

Displays in a pop-up window the topic identified by a specific context number.

1	PopupContext
2	

PopupContext Arguments

filename	appendix.hlp
context-numbe	15

7. Click **OK** to place the macro field codes in your document. The macro exists as a field code until you make your document into Help. (For more information, see "Hypertext Links and Help Macros Don't Show Up in Help" on page 178.)

Creating Startup Macros—Adding New Menus and Menu Items

In the example that follows we will create several macros that add a new menu and new menu item when a help file is loaded. In particular, we will add a new menu called Learn that has a menu item called Tutorial. Clicking on Tutorial will load a different help file called tutorial.hlp and display the contents topic.

In this example we will use "nested" macros. You can have one macro use up to two more additional macros. This nesting capability lets you combine the features of macros. In this example, two macros are used: InsertItem and JumpID. JumpID is the nested macro.

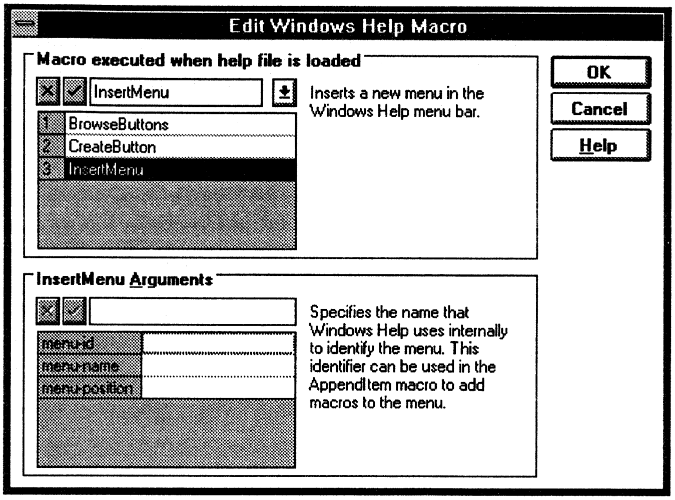
To Create a Startup Macro that Adds a New Menu and New Menu Item

1. From the **Format** menu, choose **Set Project Options**. The Set Project Options dialog box will appear.
2. Click the **Help Options** button. The extended Set Project Options dialog box will appear.
3. Click the **Startup Macros** button. The Edit Windows Help Macro dialog box will appear.

Doc-To-Help uses the CreateButton macro to add the Glossary button.

4. Your project may already have at two Startup macros listed, BrowseButtons and CreateButton. If this is the case, click in the first empty cell below these macros.

5. The macro for creating a new menu is InsertMenu. From the drop-down list box, select **InsertMenu**. Your screen should look like the one shown below.



Macro executed when help file is loaded: InsertMenu

Inserts a new menu in the Windows Help menu bar.

1	BrowseButtons
2	CreateButton
3	InsertMenu

menu-id	
menu-name	
menu-position	

Specifies the name that Windows Help uses internally to identify the menu. This identifier can be used in the AppendItem macro to add macros to the menu.

You can also add items to Help's Standard menus. See "Help Menus and Menu Items" on page 168.

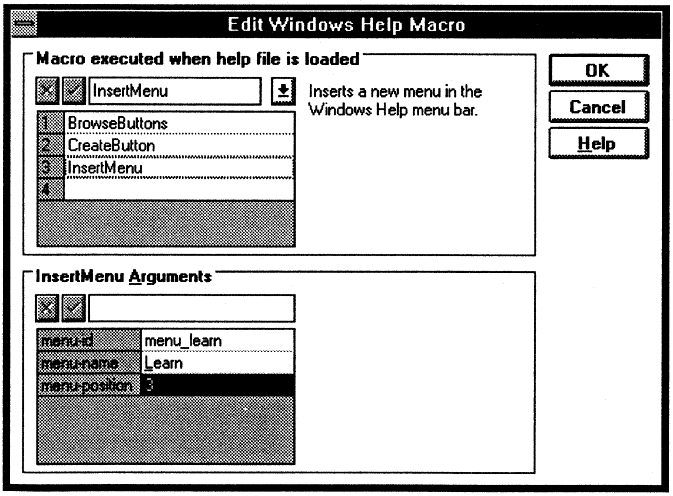
6. In the menu-id box, type **menu_learn**. This is the name you use to identify the menu or one of the names Help uses to identify its standard menus.

7. In the menu-name box, type **&Learn**. This is the text that you want to appear on the menu.

8. In the menu-position box, type **3**. This specifies the position on the menu bar of the new menu name. Positions are numbered from left to right, with the leftmost being position 0.

9. Click the InsertMenu Arguments checkmark button.

10. Click the Macro executed when... checkmark button. Your screen should look like the one shown below.



Macro executed when help file is loaded: InsertMenu

Inserts a new menu in the Windows Help menu bar.

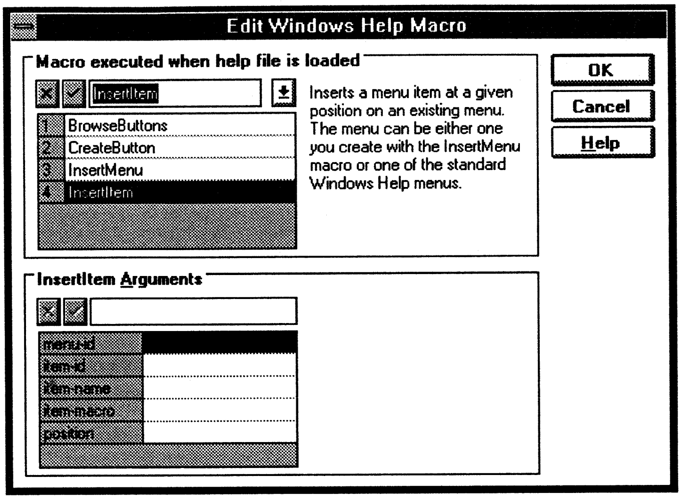
1	BrowseButtons
2	CreateButton
3	InsertMenu
4	

menu-id	menu_learn
menu-name	Learn
menu-position	3

At this point we have added a new menu called Learn which will appear to the right of the Bookmark menu. In the next series of steps we will add the menu item "Tutorial" that when clicked will take the user to the contents topic of a help file named tutorial.hlp.

11. Click in the first blank cell, which in this example is cell 4.

12. The macro for creating a menu item is InsertItem. From the drop-down list box, select **InsertItem**. Your screen should look like the one shown below.



InsertItem

1 BrowseButtons
2 CreateButton
3 InsertMenu
4 InsertItem

Inserts a menu item at a given position on an existing menu. The menu can be either one you create with the InsertMenu macro or one of the standard Windows Help menus.

13. In the menu-id box, type **menu_learn**. This is the name you use in the InsertMenu macro to identify the menu or one of the names Help uses to identify its standard menus.

14. In the item-id box, type **item_tutorial**. This is the label Help uses internally to identify the menu item. You can use any label, but it must be different from any other item-id you have used.

15. In the item-name box, type **T&utorial**. This will place the word Tutorial on the Learn menu and make the letter "u" its keyboard shortcut.

You can access the … button with the keyboard by holding down the Alt key and pressing period (.).

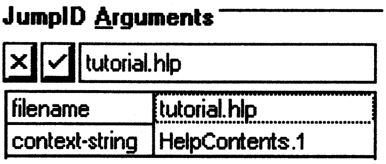
16. Click in the item-macro box. The … button appears in the InsertItem Arguments section. This button guides you in adding a nested macro.

17. Click the … button. A second Help Macro dialog box appears. You have to specify which macro you want executed when the user clicks on the menu item Tutorial. In this example, when the menu item is clicked, the user jumps to a file named tutorial.hlp. This second dialog box allows you to specify the jump macro and arguments you want executed when the user clicks the Tutorial menu item.

18. From the drop-down list box, select **JumpID**. The JumpID Arguments section appears in the dialog box.

If tutorial.hlp is not in the path or the same directory as the main help file, you should include the full path. In this case you'll need to specify four backslashes in the path because the JumpID macro is embedded within the InsertItem macro.

19. In the filename box, type **tutorial.hlp**. This is the file that will appear when the user clicks the menu item.
20. In the context-string box, type **HelpContents.1**. This is the context string of the Contents topic in Tutorial.hlp. This section of the dialog box should now look like this:

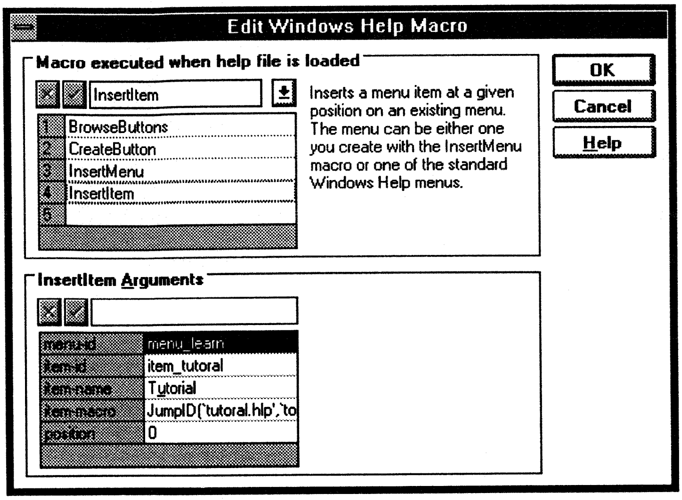


JumpID Arguments

☒ ☑ tutorial.hlp

filename	tutorial.hlp
context-string	HelpContents.1

21. Click the checkmark button in the "JumpID" Arguments section. This completes the arguments.
22. Click the checkmark button in the "Macro executed from menu item" section. This completes the macro definition for JumpID.
23. Click **OK**. The first Help Macro dialog box reappears and the JumpID macro appears in the item-macro box in the InsertItem Arguments section.
24. In the position box, type **0**. This is the number specifying where the new menu item will appear. The number must be an integer. The first item on a menu is represented by 0, the second item is 1, and so forth. So in this example the word Tutorial will be the first item in the Learn menu.
25. Click the checkmark button in the "InsertItem" Arguments section. This completes the arguments.
26. Click the checkmark button in the "Macro executed when help file is loaded" section. This completes the macro definition. Your screen should look like the one shown below.



Inserts a menu item at a given position on an existing menu. The menu can be either one you create with the InsertMenu macro or one of the standard Windows Help menus.

menu-id	menu_learn
item-id	item_tutoral
item-name	Tutorial
item-macro	JumpID(`tutoral.hlp',`to
position	0

27. Click **OK.** Doc-To-Help adds the appropriate macro to your Help Project File (HPJ File).

Creating Startup Macros—Creating a Button

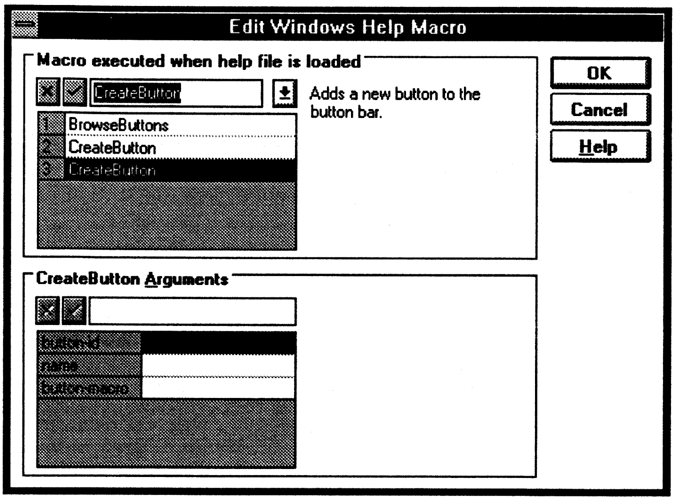
You can use start-up macros to create a button that appears every time a user opens the Help file. The following example creates a button labeled Addresses that appears on the button bar every time the Help file is open. When the button is clicked, it takes the user to the Contents topic of a separate Help file named Contacts.

To Create a Startup Macro that Adds a Button

1. From the **Format** menu, choose **Set Project Options**. The Set Project Options dialog box will appear.
2. Click the **Help Options** button. The extended Set Project Options dialog box will appear.
3. Click the **Startup Macros** button. The Edit Windows Help Macro dialog box will appear.
4. Your project may already have two Startup macros listed, BrowseButtons and CreateButton. If this is the case, click in the first empty cell below these macros.
5. The macro for creating a button is CreateButton. From the drop-down list box, select **CreateButton**. Your screen should look like the one shown below.

Doc-To-Help uses the CreateButton macro to add the Glossary button.

You can also use the CreateButton macro shortcut, CB.



Edit Windows Help Macro

Macro executed when help file is loaded

CreateButton

Adds a new button to the button bar.

1	BrowseButtons
2	CreateButton
3	CreateButton

CreateButton Arguments

button-id	
name	
button-macro	

6. In the button-id box, type **btn_addresses**. This is the unique name you use to identify the button you are creating.

7. In the name box, type **&Addresses**. This is the text that you want to appear in the button.

8. Click in the button-macro box. The … button appears in the CreateButton Arguments section. This button guides you in adding a nested macro.

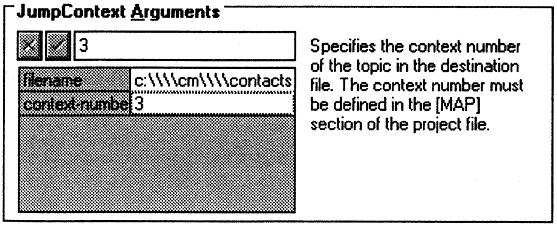
9. Click the … button. A second Help Macro dialog box appears. You have to specify which macro you want executed when the user clicks on the Addresses button. In this example, when the menu item is clicked, the user jumps to a file named Contacts. This second dialog box allows you to specify the jump macro and arguments you want executed when the user clicks the Addresses button.

10. From the drop-down list box, select **JumpContext**. The JumpContext Arguments section appears in the dialog box.

Yes, you really need four backslashes. If the file you want to jump to is not in your path, or is not in the same directory as the file you are jumping from, you must specify the entire path. In this case you need four backslashes because the JumpContext macro is embedded within the CreateButton macro.

11. In the filename box, type **c:\\\\cm\\\\contacts.hlp**. This is the file that will appear when the user clicks on the button.

12. In the context-number box, type the topic number you want to jump to, in this case **1**. This section of the dialog box should now look like this:



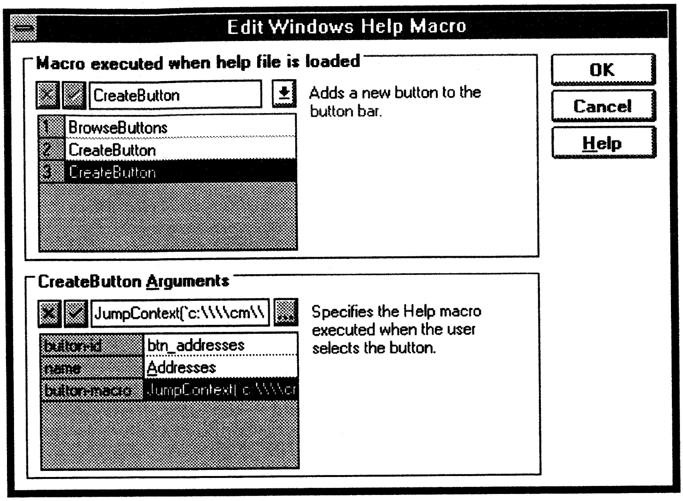
JumpContext Arguments

3

filename	c:\\\\cm\\\\contacts
context-numbe	3

Specifies the context number of the topic in the destination file. The context number must be defined in the [MAP] section of the project file.

13. Click on the checkmark in the "Macro executed from button" section and click **OK**. This will return you to the first dialog box. Your screen should look like the one shown below.



Edit Windows Help Macro

Macro executed when help file is loaded

CreateButton

Adds a new button to the button bar.

1 BrowseButtons
2 CreateButton
3 CreateButton

CreateButton Arguments

JumpContext(`c:\\\\cm\\

Specifies the Help macro executed when the user selects the button.

button-id	btn_addresses
name	Addresses
button-macro	JumpContext(`c:\\\\cr

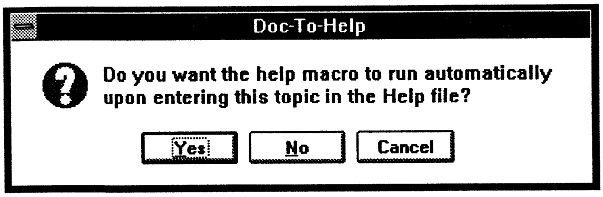
14. Click **OK**.

Help Topic Macros—Creating a Button When a Topic Is Displayed

You can create Windows Help macros that start whenever the user enters a certain topic. This is useful for customizing topics with buttons, menu items, and markers. The following example uses "nested" macros. This nesting capability lets you combine the features of macros. In the example two macros are used: CreateButton and JumpID. JumpID is the nested macro. When the user enters a particular Help topic, a button labeled "Tutorial" will appear. Clicking the Tutorial button jumps the user to the contents screen of a separate Help file named TUTORIAL.HLP.

To Create a Button upon Entering a Topic

1. Click inside the heading that begins the topic.
    - If you highlight text in the heading instead of placing an insertion point, Doc-To-Help will display the following message:

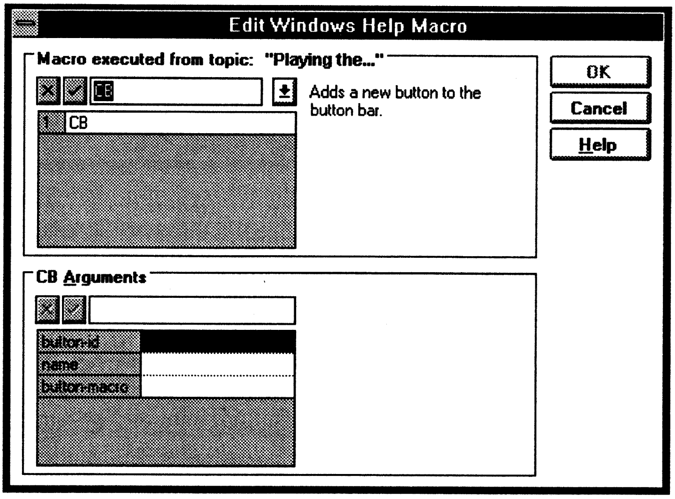


Doc-To-Help

Do you want the help macro to run automatically upon entering this topic in the Help file?

- Since you want the macro to be executed upon entering the topic, click **Yes**. If you click No, the highlighted heading text will become a Hotspot macro.

2. From the Insert menu, choose **Help Macro**. The Edit Windows Help Macro dialog box appears.
3. The macro for creating a button is CreateButton or CB. (CB is a valid abbreviation for the CreateButton macro.) From the drop-down list box, select **CreateButton** or **CB**. Your screen should look like the one shown below.

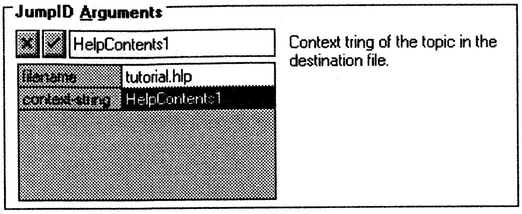


Adds a new button to the button bar.

4. In the button-id box, type **btn_tutorial**. This is the label the Windows Help compiler uses internally to identify the new button. You can use any label, but it must be different from any other button-id you have used.
5. In the name box, type **&Tutorial**. This is the text that will appear on the button.
6. Click in the button-macro box. The … button appears in the CreateButton Arguments section. This button guides you in adding a nested macro.
7. Click the … button. A second Help Macro dialog box appears. From this second dialog box you can specify the jump macro and arguments you want executed when the user clicks the Tutorial button.
8. From the drop-down list box, select **JumpID**. The JumpID Arguments section appears in the dialog box.

If tutorial.hlp is not on the path or in the same directory as the main help file, you should include the full path. In this case you'll need to specify four backslashes in the path because the JumpID macro is embedded within the InsertItem macro.

9. In the filename box, type **tutorial.hlp**.
10. In the context-string box, type **HelpContents.1**. This is the context string of the Contents topic in Tutorial.hlp. The JumpID Arguments section should look like this.



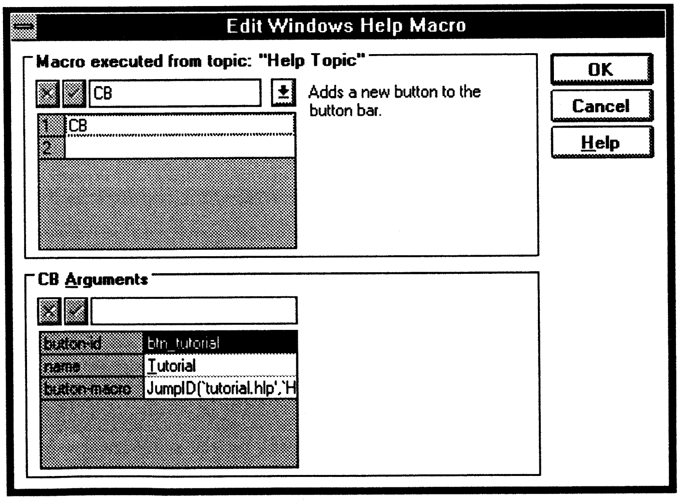
JumpID Arguments

HelpContents1

Context tring of the topic in the destination file.

filename	tutorial.hlp
context-string	HelpContents1

11. Click the checkmark button in the "JumpID" Arguments section. This completes the arguments.
12. Click the checkmark button in the "Macro executed from button" section. This completes the macro definition for JumpID.
13. Click **OK**. The first Help Macro dialog box reappears and the JumpID macro appears in the button-macro box in the CreateButton Arguments section.
14. Click the checkmark button in the "CB" Arguments section. This completes the arguments.
15. Click the checkmark button in the "Macro executed from topic" section. This completes the macro definition for CreateButton. Your screen should look like the one shown below.



Edit Windows Help Macro

Macro executed from topic: "Help Topic"

CB

Adds a new button to the button bar.

1	CB
2	

CB Arguments

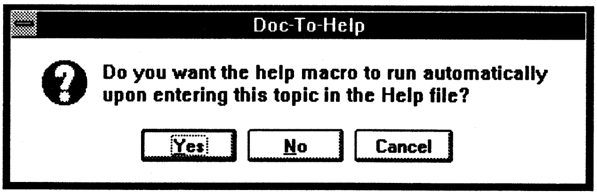
button-id	btn_tutorial
name	Tutorial
button-macro	JumpID(`tutorial.hlp',`H

16. Click **OK** to place the macro field codes in your document. The macro exists as a field code until you make your document into Help. (For more information, see "Hypertext Links and Help Macros Don't Show Up in Help" on page 178.)

Help Topic Macros—Removing a Button When a Topic Is Displayed

You can use help topic macros to remove a button from a particular topic. This example removes a button that was previously created with a button-id of btn_glossary. The button is removed as soon as the user enters the topic that contains the macro.

1. Click inside the heading that begins the topic.
   - If you highlight text in the heading instead of placing an insertion point, Doc-To-Help will display the following message:

     Do you want the help macro to run automatically upon entering this topic in the Help file?

   - Since you want the macro to be executed upon entering the topic, click **Yes**. If you click No, the highlighted heading text will become a Hotspot macro.
2. From the Insert menu, choose **Help Macro**. The Edit Windows Help Macro dialog box appears.
3. The macro for eliminating a button is DestroyButton or DB. (DB is a valid abbreviation for the DestroyButton macro.) From the drop-down list box, select **DestroyButton** or **DB**. The DestroyButton (or DB) Arguments section appears in the dialog box.
4. In the button-id box, type the id of the button you want to remove, in this case **btn_glossary.**
5. Click the checkmark button in the "DestroyButton" Arguments section. This completes the arguments.
6. Click the checkmark button in the "Macro executed from topic" section. This completes the macro definition for DestroyButton.
7. Click **OK** to place the macro field codes in your document. The macro exists as a field code until you make your document into Help. (For more information, see "Hypertext Links and Help Macros Don't Show Up in Help" on page 178.)

Note: This macro cannot be used to delete any of the standard Help buttons (Contents, Search, Back, or History). Help also ignores this macro if it is executed in a secondary window.

Editing and Deleting Macros

Editing a Hotspot Macro

To edit a Hotspot macro, highlight the text or graphic that when clicked will execute the macro. Choose **Help Macro** from the **Insert** menu.

The Edit Windows Help Macro dialog box will appear with the current macro settings. Edit the macro(s) as needed and click **OK**. You can remove a cell entry by selecting the cell and pressing the **Delete** key.

Editing a Help Topic Macro

To edit a Help Topic macro, click inside the heading that begins the topic and choose **Help Macro** from the **Insert** menu.

The Edit Windows Help Macro dialog box will appear with the current macro settings. Edit the macro(s) as needed and click **OK**. You can remove a cell entry by selecting the cell and pressing the **Delete** key.

Deleting a Hotspot or Help Topic Macro

When you use the **Insert Help Macro** command in your manual to create a hypertext link or macro in the Help file, Doc-To-Help inserts a special **\Relate** field to specify the button text and target of the jump.

An example of a relate field is shown below.

{\relate "ButtonText" \D2HHelpMacro MacroText}

If you want to remove a help macro or hypertext link, do the following:

1. Open the document containing the field you want to delete.
2. Make sure that **Field Codes** are turned on in the **View** menu. There should be a check mark next to the **Field Codes** command.
3. Select the invalid \Relate field, and delete it.

For more information on Relate fields, see “Hypertext Links and Help Macros Don’t Show Up in Help” on page 178.

Editing or Deleting a Startup Macro

1. From the **Format** menu, choose **Set Project Options**. The Set Project Options dialog box will appear.
2. Click the **Help Options** button. The extended Set Project Options dialog box will appear.
3. Click the **Startup Macros** button. The Edit Windows Help Macro dialog box will appear.
4. Edit the macros as needed. To delete the contents of a cell, select the cell and press the **Delete** key.

Adding Sound to Your Help Files

As we mentioned earlier, you can extend the basic functionality of WINHELP by registering external Dynamic Link Library (DLL) functions and subroutines and running these routines from a Help macro. Microsoft Windows 3.1 ships with a DLL called MMSYSTEM.DLL that contains a routine called sndPlaySound that will play .WAV files.

Where Do I Find Sounds? How Can I Hear Sounds?

Windows ships with a few simple sounds (TADA.WAV, CHIMES.WAV) etc. You can also find commercial and shareware .WAV libraries, or can record and create your own sounds using the software that comes with sound boards.

To hear "decent" sounds you'll need a sound board and speakers. If you're willing to put up with scratchy but serviceable sound, you should get a hold of a file called SPEAKER.DRV which allows you to play sounds through your computer's built-in speaker without purchasing additional hardware. SPEAKER.DRV is available on many bulletin boards, including CompuServe.

Registering the Routine

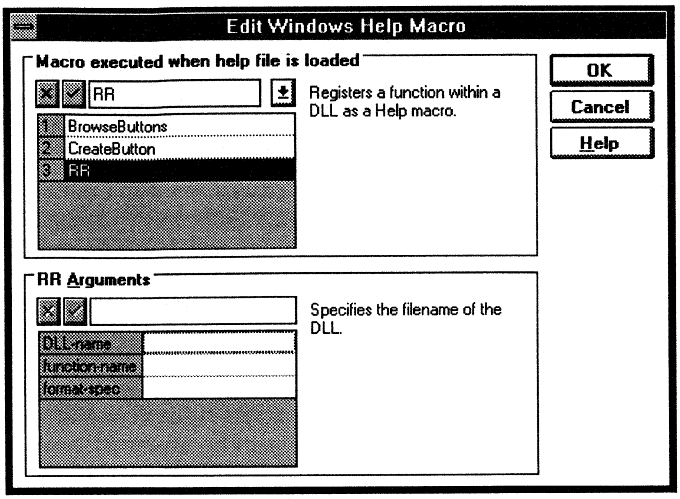
The first thing you'll need to do is add a Startup macro that registers the routine you'll be calling when you play a sound. Since the routine is registered through a Startup macro, the routine will be available as soon as the help file loads.

To Register the Routine that Plays Sounds

1. From the **Format** menu, choose **Set Project Options**. The Set Project Options dialog box will appear.
2. Click the **Help Options** button. The extended Set Project Options dialog box will appear.
3. Click the **Startup Macros** button. The Edit Windows Help Macro dialog box will appear.
4. Your project may already have two Startup macros listed, BrowseButtons and CreateButton. If this is the case, click in the first empty cell below these macros.

Doc-To-Help uses the CreateButton macro to add the Glossary button.

5. The macro for creating a new menu is RegisterRoutine or RR. From the drop-down list box, select **RR**. The RR Arguments group box will appear. Your screen should look like the one shown below.



Macro executed when help file is loaded

RR

Registers a function within a DLL as a Help macro.

1	BrowseButtons
2	CreateButton
3	RR

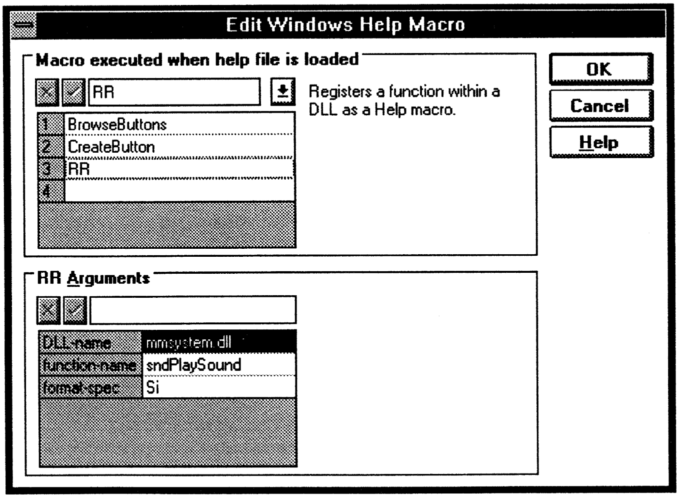
RR Arguments

Specifies the filename of the DLL.

DLL-name	
function-name	
format-spec	

6. In the DLL-name box, type **mmsystem.dll**.
7. In the function-name box, type **sndPlaySound**.
8. In the format-spec box, type **Si**.
9. Click the "RR Arguments" checkmark button.
10. Click the "Macro executed when help file is loaded" checkmark button. Your screen should look like the one shown below.

*In the format-spec edit box you enter a string that specifies the types of parameters that are passed to the funtion where u=unsigned short, U=unsigned long, i=short int, I=int, s=near char *, S=far char *, and v=void. Specifying Si indicates that the first parameter is a string and the second is a short integer.*



Macro executed when help file is loaded

RR

Registers a function within a DLL as a Help macro.

1	BrowseButtons
2	CreateButton
3	RR
4	

RR Arguments

DLL-name	mmsystem.dll
function-name	sndPlaySound
format-spec	Si

11. Click **OK**.

Please note that the sndPlaySound function has been added to the Macro drop down list and can now be called like any other Help macro.

You can also add a Startup macro that plays a sound by inserting the newly declared sndPlaySound function in any blank cell below the RR macro. We'll discuss how to create a macro that plays a sound in the next section.

Creating a Hotspot that Plays a Sound

The steps needed to play a sound are essentially the same for Hotspot, Help Topic and Startup macros. What follows is a discussion of how to make it so that clicking on text or a graphic plays a sound.

In all of these cases it is essential that the sndPlaySound routine be registered before you attempt to play a sound.

To Create a Hotspot that Plays a Sound

1. Highlight the text or graphic that when clicked will play a sound.
2. From the **Insert** menu, choose **Help Macro**. The Edit Windows Help Macro dialog box will appear.
3. From the Macro drop-down list, choose **sndPlaySound**. The sndPlaySound Arguments box will appear.
4. In the Far String box, type the name of the sound file you want to play. The file should either be in the same directory as the help file or on the path; otherwise you will need to specify the drive name and path name as well as the file name.
5. In the Int box, type **0**.
6. Click on the sndPlaySound Arguments checkmark button.
7. Click on the "Macro executed from text" checkmark button.
8. Click **OK**.

The 0 indicates that you want the entire sound to play before returning control back to the user. Entering a 1 will play the sound in the background. For more information, see the Multimedia Programmer's Reference.

Help Macro Reference

For a complete discussion of Windows Help Macros and their uses, see Chapter Ten, "Working with Help Macros" in *Developing Online Help for Windows*, by Scott Boggan, David Farkas and Joe Welinske. The book is published by Sams Publishing, ISBN 0-672-30230-6.

The Standard Button Bar

Help provides six buttons for use with Help files. The standard buttons provide access to six commonly used Help macros. Four of the buttons are required, and two are optional buttons in your Help file.

You can access the standard button functionality if you use the specified button IDs. The standard buttons have the following IDs, macros, and functions.

Button	ID	Macro Name	Description
Contents	btn_contents	Contents	Displays the Contents topic, which is the first topic in the Help file.
Index	btn_search	Search	Displays the Search dialog box.
Go Back	btn_back	Back	Jumps to the last topic the user displayed in the main window.
History	btn_history	History	Displays the History window.
<< (Browse Previous)	btn_previous	Prev	Jumps to the previous topic in the browse sequence.
>> (Browse Next)	btn_next	Next	Jumps to the next topic in the browse sequence.

In certain situations, Help automatically disables buttons with the following button IDs.

Button ID	Disabled when
btn_back	Using the Go Back button or the history list, the user returns to the first topic in the history list.
btn_previous	Help is displaying the first topic in the browse sequence.
btn_next	Help is displaying the last topic in the browse sequence.

This behavior provides useful feedback to the users. If you define custom Previous and Next buttons without using the button IDs listed in the preceding table, Help cannot disable the buttons in the situations described above. To define these buttons using custom paging devices, be sure to use the specified button IDs in your CreateButton or InsertButton macros.

Customizable Buttons

These macros allow you to add or remove, enable or disable buttons. You can also specify another macro to work when a certain button is clicked.

CreateButton
DestroyButton
DisableButton
EnableButton
SetContents

Help Menus and Menu Items

Windows Help provides a standard menu bar for use with Help files. You can access the standard menu functionality with macros as long as you know the specified menu IDs. The standard help menu has the following menus, menu item, IDs and macro assignments.

Menu	ID	Macro	Description
File menu	mnu_file	N/A	Macros cannot be run directly from the menu bar.
Edit	mnu_edit	N/A	""
Bookmark	mnu_bookmark	N/A	""
Help	mnu_help	N/A	""

Menu Item	ID	Macro	Description
File Open	mnu_open	FileOpen	Displays the Open dialog box.
File Print	mnu_print	Print	Prints the topic displayed in the main window.
File Print Setup	mnu_psetup	PrinterSetup	Displays the Print Setup dialog box.
File Exit	mnu_exit	Exit	Closes Help.
Edit Copy	mnu_copy	CopyDialog	Displays the Copy dialog box.
Edit Annotate	mnu_annotate	Annotate	Displays the Annotate dialog box.
Bookmark Define	mnu_bkdefine	BookmarkDefine	Displays the Bookmark Define dialog box.
Bookmark list	None2	None2	Lists the first nine bookmarks defined in the Help file. These items are displayed only if bookmarks are defined.
Bookmark More	mnu_bkmore	BookmarkMore	Displays the Bookmark dialog box for Help files that have more than nine bookmarks defined. This menu item is displayed only if ten or more bookmarks are defined.
Help How To Use Help	mnu_helpon	HelpOn	Displays the How To Use Help file in a new Help window.
Help Always On Top	mnu_ontop	HelpOnTop	Displays all Help windows on top of other application windows.
Help About	mnu_about	About	Displays the About dialog box.

You can add or remove menus and menu items, and specify a macro to execute when a certain item is selected. You can also add or remove check marks beside menu items. Menu items can include keyboard accelerators or mnemonic characters.

AppendItem
ChangeItemBinding
CheckItem
DeleteItem
DisableItem
EnableItem
InsertItem
InsertMenu
SetHelpOnFile
UncheckItem

Extensions

You can launch Windows applications and link to DLLs using these the two macros.

ExecProgram
RegisterRoutine

Help Windows

With these macros you can modify the size and appearance of secondary windows. If you only work with the main windows, you can set those characteristics through Set Project Options.

CloseWindow
FocusWindow
PositionWindow

Help Markers

You can add conditional abilities to your macros with the assistance of the marker macros. These macros add If/Then capabilities to your placement of menus and buttons.

DeleteMark
GotoMark
IfThen
IfThenElse
IsMark
Not
SaveMark

Keyboard Accelerators

You can assign short-cut keys to certain macros. This is useful in secondary windows where you cannot enable buttons or menus. An accelerator key could be assigned to the Print macro so that the user could print a topic in a secondary window.

AddAccelerator
RemoveAccelerator

Jump/Popup Topics

These macros allow you to jump to other topics or to display topics as popup definitions. The usefulness of these macros is somewhat limited in that the Doc-To-Help method of inserting a Hypertext Jump is easier. The macros are useful when you want to jump to a topic in another file or display a popup from another file.

PopupContext
PopupId
JumpContents
JumpContext
JumpHelpOn
JumpId
JumpKeyword

Macro Definitions

Following is an alphabetical listing of all the Windows Help Macros. When you use the Doc-To-Help Macro Editor, you'll see an explanation for each macro when you select that macro from a drop-down list.

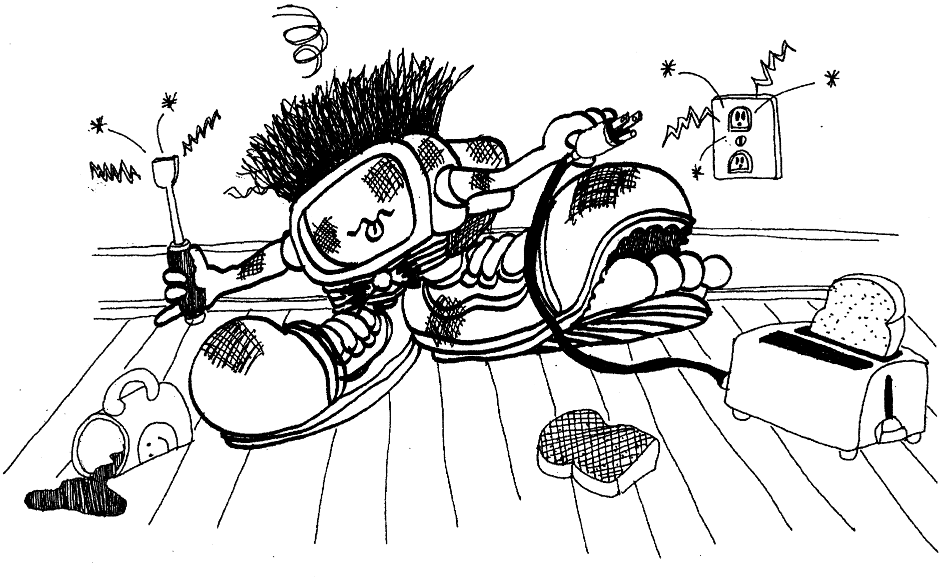
Macro Name	Description
About	Displays the About dialog box.
AddAccelerator (AA)	Assigns a Help macro to an accelerator key (or key combination) so that the macro is run when the user presses the accelerator key(s).
Annotate	Displays the Annotation dialog box.
AppendItem	Appends a menu item to the end of a menu you create with the InsertMenu macro.
Back	Displays the previous topic in the Back list. The Back list includes the last 40 topics the user has displayed since starting WinHelp.
BookmarkDefine	Displays the Define dialog from the Bookmark menu.
BookmarkMore	Displays the More dialog from the Bookmark menu. The More command appears on the Bookmark menu if the menu lists more than nine bookmarks.
BrowseButtons	Adds browse buttons to the button bar.
ChangeButtonBinding (CBB)	Assigns a Help macro to a Help button.
ChangeItemBinding (CIB)	Assigns a Help macro to an item previously added to a Windows Help menu using the AppendItem macro.
CheckItem (CI)	Places a check mark beside a menu item.
CloseWindow	Closes either a secondary window or the main Help window.
Contents	Displays the Contents topic in the current Help file.
CopyDialog	Displays the Copy dialog from the Edit menu.
CopyTopic	Copies all the text in the currently displayed topic to the Clipboard.
CreateButton (CB)	Adds a new button to the button bar.
DeleteItem	Removes a menu item that was added by using the AppendItem macro.
DeleteMark	Removes a text marker added with the SaveMark macro.

Macro Name	Description
DestroyButton (DB)	Removes a button added with the CreateButton macro.
DisableButton	Grays out a button added with the CreateButton macro. This button cannot be used in the topic until an EnableButton macro is executed.
DisableItem (DI)	Grays out a menu item added with the AppendItem macro. The menu item cannot be used in the topic until an EnableItem macro is executed.
EnableButton (EB)	Re-enables a button disabled with the DisableButton macro.
EnableItem (EI)	Re-enables a menu item disabled with the DisableItem macro.
ExecProgram (EP)	Executes a Windows application.
Exit	Exits the Windows Help application. It has the same effect as selecting Exit from the File menu.
FileOpen	Displays the Open dialog box from the File menu.
FocusWindow	Changes the focus to the specified window, either the main Help window or a secondary window.
GotoMark	Jumps to a marker set with the SaveMark macro.
HelpOn	Displays the Help file for the Windows Help application. The macro carries out the same action as choosing the How to Use Help command on the Help menu.
History	Displays the history list, which shows the last 40 topics the user has viewed since opening a Help file in Windows Help. It has the same effect as choosing the History button.
IfThen	Executes a Help macro if a given marker exists, using the IsMark macro to make the test. The result of the test can be reversed by enclosing the IsMark macro within the Not macro.
IfThenElse	Executes one of two Help macros depending on whether or not a marker exists, as tested by the IsMark macro. The result of the test can be reversed by enclosing the IsMark Macro within the Not macro.
InsertItem	Inserts a menu item at a given position on an existing menu. The menu can be either one you create with the InsertMenu macro or one of the standard Windows Help menus.

Macro Name	Description
InsertMenu	Inserts a new menu in the Windows Help menu bar.
IsMark	Used with IfThen and IFThenElse to determine if a text marker perviously created with the SaveMark macro exists.
JumpContents	Jumps to the Contents topic of a specified Help file.
JumpContext (JC)	Jumps to a topic identified by a context number. The context is identified by an entry in the [MAP] section of the .HPJ file.
JumpHelpOn	Jumps to the Contents topic of the How to Use Help file. The How To Use Help file is either the default WINHELP.HLP file shipped with Windows 3.1 or the Help file designated by the SetHelpOnFile macro.
JumpId (JI)	Jumps to the topic with the specified context string in the Help file.
JumpKeyword (JK)	Loads the indicated Help file, searches through the K keyword table, and displays the first topic containing the index keyword specified in the macro.
Next	Displays the next topic in the browse sequence for the Help file.
NotIsMark	Used with IfThen and IFThenElse to determine if a text marker perviously created with the SaveMark macro does not exist.
PopupContext (PC)	Displays in a popup window the topic identified by a specific context number.
PopupId (PI)	Displays a topic from a specified file in a popup window.
PositionWindow	Sets the size and position of a window.
Prev	Displays the previous topic in the browse sequence for the Help file. If the currently displayed topic is the first topic of a browse sequence, this macro does nothing.
Print	Sends the currently displayed topic to the printer. It should be used only to print topics in windows other than the main Help window (for example, topics in a secondary window).
PrinterSetup	Displays the Printer Setup dialog box from the File menu.

Macro Name	Description
RegisterRoutine (RR)	Registers a function within a DLL as a Help macro.
SaveMark	Saves the location of the currently displayed topic and file and associates a text marker with that location. The GotoMark macro can then be used to jump to this location.
Search	Displays the dialog for the Search button, which allows users to search for topics using keywords defined by the K footnote character.
SetContents	Designates a specific topic as the Contents topic in the specified Help file.
SetHelpOnFile	Designates the specific Help file that replaces WINHELP.HLP, the default Using Help file in the Windows environment.
UncheckItem (UI)	Removes a check mark besides a menu item.

Troubleshooting



Troubleshooting

Overview

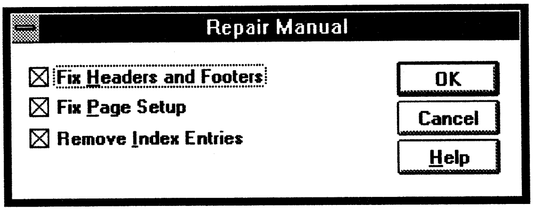
Every now and then you may run into trouble with improperly formatted documents, memory errors while converting your manuals into Help, or working with very large manuals. This section explains how to correct many common problems.

How Tools Repair Manual Corrects Documents

The **Tools Repair Manual** command can be used to fix the following problems with your manual:

- Headers and Footers aren't formatted properly.
- Page Layout doesn't match the original template or the selection in **Set Project Options**.
- Extraneous index entries need to be removed.
- RD fields in a multifile document are incorrect.

Choosing **Tools Repair Manual** from a single-file project will display the following dialog box, as shown below.



Repair Manual

☒ Fix Headers and Footers

☒ Fix Page Setup

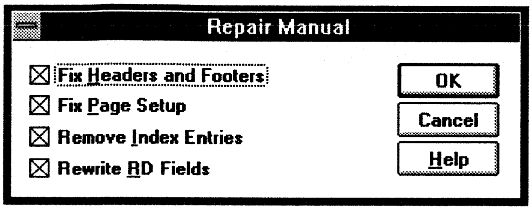
☒ Remove Index Entries

Choosing **Fix Headers and Footers** will reset the headers and footers in your manual to match the headers and footers in the attached template.

Choosing **Fix Page Setup** will match the page setup in the manual to the default in the attached template.

Choosing **Remove Index Entries** will remove Word for Windows index fields and Doc-To-Help temporary index codes from the manual.

If you are repairing a multifile project and the master document is active, the **Tools Repair Manual** dialog box will have an additional checkbox for repairing RD fields as shown below.



☒ Fix Headers and Footers
☒ Fix Page Setup
☒ Remove Index Entries
☒ Rewrite RD Fields

Choosing **Rewrite RD Fields** will delete existing RD fields and update them to reference the correct inside chapter files. RD fields are used to compile the Index and Table of Contents in your master document.

Hypertext Links and Help Macros Don't Show Up in Help

When you use either the **Insert Hypertext Link** command or the **Insert Help Macro** command in your manual to create a hypertext link or macro in the Help file, Doc-To-Help inserts a special **\Relate** field to specify the button text and target of the jump. If the button text is changed or deleted, or if the bookmark specifying the target is deleted, several problems may occur:

1. During the Create Hypertext Links and Macros phase of making your manual into Help, Doc-To-Help beeps and displays an error message in the status bar.
2. A word or phrase intended as button text appears instead as static text in the Help file.
3. Button text is missing entirely from the Help file.
4. Macros do not run correctly.

Repairing a Bad Help Macro or Hypertext Link

*You can use **Doc-To-Help Diagnostics** to search for all invalid Hypertext Links and Help Macros topics in your manual. (See "Tools Doc-To-Help Diagnostics" on page 236.)*

For help macros you should highlight the button text which when clicked is supposed to invoke the macro, and choose **Insert Help Macro**. The Doc-To-Help Macro Editor will allow you to edit the \Relate field without getting your hands dirty. For Hypertext Links you should highlight the button text that is supposed to produce a jump or a popup, then choose **Insert Hypertext Link**. Doc-To-Help will tell you that a hypertext link already exists and ask you if you want to replace it. Since the link is not working correctly, click **Yes**.

Deleting a Help Macro or Hypertext Link

If you want to remove a help macro or hypertext link, do the following:

1. Open the source document containing the offending field.
2. Make sure that **Field Codes** are turned on in the **View** menu. There should be a check mark next to the **Field Codes** command.
3. Select the invalid \Relate field, and delete it.

Relate Field Syntax

Jump

{\Relate "File!BookmarkNum", "ButtonText"}

The first parameter is the name of the file containing the target topic, followed by an exclamation point, and then the name of a bookmark that references the topic to which you are linking.

The second parameter indicates the button text for the jump, which must immediately precede the \Relate field. If there is no button text, the title of the target topic will be inserted in the help file at the location of the field, and used as the button text. If the hot spot is a graphic, the button text argument will be "^1".

Popup

{\relate "File!BookmarkName", "ButtonText" \D2HPopup}

Secondary Window

{\relate "File!BookmarkName", "ButtonText" \D2HWindow WindowName}

Help Macro

{\relate "ButtonText" \D2HHelpMacro MacroText}

In the case of a Help Macro relate field, if the ButtonText argument is omitted the macro will execute automatically upon entering the topic where the help macro relate field appears, provided that the field is in a heading paragraph. Otherwise the field will be ignored.

Paragraph Doesn't Wrap in Help Screens

This problem occurs when **Keep Lines Together** was applied as explicit formatting to a paragraph in the document. The Help compiler interprets a paragraph with this formatting applied to it as one long line. Because Doc-To-Help does not strip explicit formatting when a document is made into Help you need to either remove the formatting before making the document into Help or only apply the formatting through a style. (See "Redefining Styles in Help" on page 107.)

You can also remove the formatting from within the RTF file itself. You can use the **Paragraph** button in the **Edit Find** dialog box to search for paragraphs with **Keep Lines Together**. Then use **Format Paragraph** to remove it. You'll need to recompile the Help file when you're finished.

Please note that table text will not wrap if the help screen is not wide enough to accommodate the entire table.

Graphics Aren't Positioned Correctly in Help Screens

The Microsoft Help Compiler ignores frame formatting when you convert your document into Help. If you've used frames to position graphics in your manual you will need to remove the frame and reposition graphics using paragraph formatting.

Frames are only positioned in **Page Layout View** or **Print Preview**. The easiest way to tell how graphics will be positioned in your Help file is to look at the document in **Normal View** which ignores frame formatting.

Alignment Problems in the Help File

Paragraphs that Look Fine in the Manual Are Indented in the Help File

If you examine the paragraph formatting for many the styles in the Doc-To-Help manual templates, you'll find that styles like Body, List and Heading 3 are indented from the left. If you examine these same style names in the Doc-To-Help help file template (D2H_HELP.DOT) you'll see that the styles are similar, but that paragraphs are formatted without the indent.

If you use the Doc-To-Help built-in styles, you won't run into any indent problems in the help file. If, however, you create new styles in the manual or use direct formatting instead of applying styles, text is indented in the help file. In general, we recommend that you avoid direct formatting. We do, however, encourage you to create your own styles. If you do create your own styles for the manual, make sure you create a corresponding style for help file. For example, let's say you create a style in your manual called SpecialBullets that indents text two inches from the left. When you make the manual into help, Doc-To-Help will respect the style name and preserve the style in the help file, leaving the two-inch indent. The solution is to also create a style called SpecialBullets in D2H_HELP.DOT that is the same as the manual's version, but without the two-inch indent.

Some Hanging Indents Don't Wrap Properly in the Help File

If you use either List or List2 styles to create paragraphs with hanging indents, your paragraphs will line up properly in both the manual and the help file. If you will be creating your own styles that employ a hanging indent, make sure that you place a left tab where the indent should be. While Word for Windows doesn't need this tab to correctly display the hanging indent, the Microsoft Help Compiler does.

Avoid creating bulleted and numbered lists using Body style as this style has not been set up to produce a hanging indent in the help file.

Accented Characters in Path Names

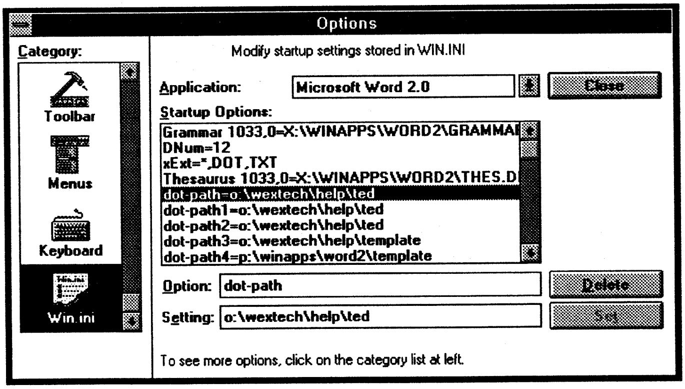
WordBasic cannot handle diacritical marks (e.g., é, ô, etc.) in path names. Make sure that all of the paths you specify for installation and any directory and file names you use in projects do not contain accented characters.

"Word Cannot Open This Document Template" Error

This error occurs when Doc-To-Help cannot attach the D2H_HELP template during **Make Into Help.** It happens if the DOT-PATH setting in the WIN.INI file is set to a directory which does not contain the Help Template.

To Correct the DOT-Path Setting

1. Determine the location of the Doc-To-Help templates on your hard disk.
2. From the **Tools** menu in Word for Windows, choose **Options**, and type a **W** to access the **WIN.INI** category.



Modify startup settings stored in WIN.INI

Application: Microsoft Word 2.0

Startup Options:

```
Grammar 1033,0=X:\WINAPPS\WORD2\GRAMMAI
DNum=12
xExt=*.DOT,TXT
Thesaurus 1033,0=X:\WINAPPS\WORD2\THES.D
dot-path=o:\wextech\help\ted
dot-path1=o:\wextech\help\ted
dot-path2=o:\wextech\help\ted
dot-path3=o:\wextech\help\template
dot-path4=p:\winapps\word2\template
```

Option: dot-path

Setting: o:\wextech\help\ted

To see more options, click on the category list at left.

3. Click in the **Option** text box and type **dot-path**.
4. Click in the **Setting** text box and type the path to the directory containing your Doc-To-Help templates.
5. Click the **Set** button and click **Close**.

Help Compiler Troubleshooting

Out of Environment Space

We recommend using Windows' built-in System Configuration Editor (SYSEDIT.EXE).

An Out of Environment Space error when attempting to compile a help file indicates that the SHELL= setting in your CONFIG.SYS file is either missing or needs to be modified. Using a text editor of your choice, modify your CONFIG.SYS file so that it contains the following entry:

```
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /E:1024 /p
```

After making this modification, you'll need to reboot your computer.

"Warning 4792 - Non-Scrolling region after scrolling region"

This warning occurs when **Keep With Next** was applied using **Format Paragraph** to one or more of the paragraphs in your manual. Keep With Next is used to keep two paragraphs from being separated by a page break. This is not necessary in the Help file because paragraphs within a topic are all part of the same screen.

Headings that are formatted with the Keep With Next attribute will stay visible at the top of a help topic even if you scroll down within the topic.

Doc-To-Help does not remove explicit paragraph formatting when it converts your document into Help. Because of this, you have to remove Keep With Next before you make the document into Help. Alternatively, you can create a style that includes Keep With Next in the manual, but whose corresponding style in the help template does not include Keep With Next. Doc-To-Help Heading 2, Heading 3 and Heading 4 styles are already defined as Keep With Next. (See "Redefining Styles in Help" on page 107.)

You can also choose to remove Keep With Next from paragraphs in the RTF file itself. The easiest way to find the paragraph(s) causing the problem is to use the **Paragraph** button in the **Edit Find** dialog box to search for paragraphs with Keep With Next applied. Remove the formatting with **Format Paragraph**. You'll need to recompile the Help file when you're finished.

"1079-Out of File Handles" when Running Help Compiler

If you get this message or an "out of memory" message when trying to run the Help compiler, you may need to increase the "Files=" component of your CONFIG.SYS file. For example, you may need to change the "Files=30" setting to "Files=40."

Likewise, if you are running Doc-To-Help on a Novell Network, you may need to increase the number of File Handles in your SHELL.CFG file.

"5059-Not Enough Memory"

If you get this message when you attempt to compile your Help Project file, try using the HCP compiler instead.

1. From the **Tools** menu, choose **Doc-To-Help Options**.
2. Change the **Help Compiler Version** to **HCP**.

"2550-Invalid Path Specified in Root Option" Warning

This warning is of no consequence whatsoever. It's simply a quirk in the Windows 3.0 Help Compiler.

"4113-Unresolved Jump" Warning

If you get this message, use the Tools Doc-To-Help Diagnostics command to check for invalid headings, invalid document structure, invalid cross references\related topics and invalid caption links.

If the diagnostics routine does not find an error, try compiling using HC31.EXE or HCP.EXE.

"1100 Cannot open file filename.rtf permission denied"

This problem usually occurs if you are trying to compile a file on a network drive. During compilation, the help compiler has a tendency to write temporary files wherever it feels like it, often trying to write these files in directories to which you do not have access rights. To force the help compiler to write these temporary files to a directory to which you do have access, do the following:

1. From within Windows, access a DOS prompt.
2. At the DOS prompt, type the following:

   SET TMP=C:\TMP

   Make sure that there is room on your C: drive and that there is a directory called \TMP. Otherwise, specify a drive and directory to which you have write privileges.
3. Type **HCP *Project*** where project is the name of the master RTF file.
4. When the help compiler is finished, type **EXIT** to return to Windows.

You can make the TMP variable setting permanent by modifying your AUTOEXEC.BAT file so that it contains the line SET TMP=C:\TMP. To edit this file, we recommend using Windows' built-in System Configuration Editor (SYSEDIT.EXE).

Large Multifile Projects

Word for Windows may run out of memory in building your project's table of contents if your multifile project is made up of more than 20 files. If you want to have more than 20 chapters, we suggest consolidating several of the chapters into a single file, rather than having a separate file for each chapter. This doesn't mean you have to reduce the number of chapters in your manual; it just means that you should save more than one chapter in a file.

Out of Memory Errors

There are several reasons for getting Out of Memory errors while making into Help. It could be the version of Word that you're using, the size of your document or the number of files that your multifile project contains. This section discusses strategies for eliminating Out of Memory errors.

Use the Latest Version of Word for Windows

Some of the memory handling problems you encounter will be alleviated by acquiring the latest version of Word for Windows. If you are using Word for Windows 2.0, we strongly encourage you to acquire version 2.0c, a free maintenance release which Microsoft made available in early 1993.

Microsoft Customer Service can be reached at (800) 426-9400.

Working with Large Projects—Determining Where the Error Occurred

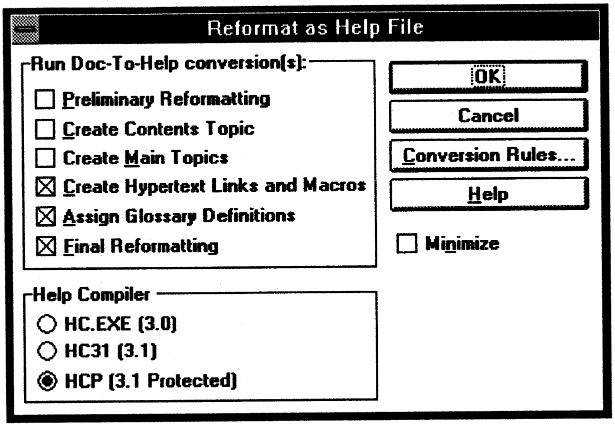
If you're encountering "Out of Memory" errors when converting a large manual into Help, this usually means that your document is too large or complex to make into Help all at once. If this is the case, you'll need to reformat the document as Help one step at a time.

There are six stages to the reformatting process. If the Out of Memory error occurred before the second stage (**Create Contents Topic**) was successfully completed, you'll need to start the process from the very beginning and make the document into Help.

The first, third, fourth, fifth and sixth stages of the **Reformat as Help** process are designed to be able to pick up where they left off, so you won't lose your work in the event of a memory error. Here's how to determine if you need to start over.

To Check What Stages Were Completed

1. From the **File** menu, choose **Save**.
2. If you're working with a multifile project, you may have more than one .RTF file open. In this case, activate each open .RTF file using the **Window** menu, and save it.
3. Quit Word for Windows.
4. Restart Word, and open your master .RTF file.
5. From the **Format** menu, choose **Reformat as Help**. The Reformat as Help File dialog box will be displayed.



Run Doc-To-Help conversion(s):

□ Preliminary Reformatting
□ Create Contents Topic
□ Create Main Topics
☒ Create Hypertext Links and Macros
☒ Assign Glossary Definitions
☒ Final Reformatting

□ Minimize

Help Compiler

○ HC.EXE (3.0)
○ HC31 (3.1)
● HCP (3.1 Protected)

6. If a step in the conversion was not completed successfully, the corresponding checkbox will be checked. In the example above, we can tell that the steps **Preliminary Reformatting, Create Contents Topic** and **Create Main Topics** were completed and that **Create Hypertext Links and Macros, Assign Glossary Definitions** and **Final Reformatting** have not been done yet.

   If the conversion has not gotten past **Preliminary Reformatting** and **Create Contents Topic** you will have to begin **Make Into Help** from the original document. Continue with the steps described in "To Start Over from the Document" below.

To Resume Formatting

1. Make sure you have completed the steps outlined in "To Check What Stages Were Completed." If the conversion has not gotten past **Preliminary Reformatting** and **Create Contents Topic** you will have to begin **Make Into Help** from the original document. Continue with the steps described in "To Start Over from the Document" below.
2. Make sure that only the checkbox for the first unfinished **Reformat as Help** stage is selected, and click **OK**. When it finishes with this stage, you can resume with the following stage, and so on until the process is complete.

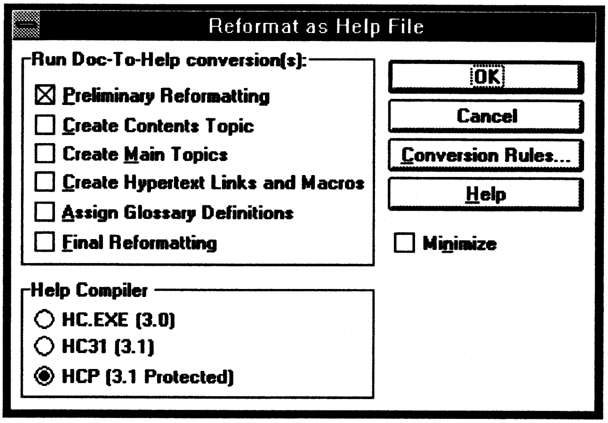
To Start Over from the Document

1. Close any open RTF files (you don't need to bother saving them).

The RTF file that was created before the out of memory error occurred will be overwritten.

2. Open your Help project's master document and choose **Make Into Help** from the **Format** menu. If you've chosen not to run **Doc-To-Help Diagnostics**, or if Doc-To-Help Diagnostics found no errors, Doc-To-Help will save your document(s) with an extension of RTF, and then display the **Reformat as Help** dialog box.

3. You'll see six check boxes for the six different stages of making your document into Help. Since you're making the document into Help one step at a time, click in the second, third, fourth, fifth and sixth checkboxes to de-select those options, and then click **OK**. Your dialog box should look like the one below.



Run Doc-To-Help conversion(s):

☒ Preliminary Reformatting
□ Create Contents Topic
□ Create Main Topics
□ Create Hypertext Links and Macros
□ Assign Glossary Definitions
□ Final Reformatting

□ Minimize

Help Compiler

○ HC.EXE (3.0)
○ HC31 (3.1)
● HCP (3.1 Protected)

4. When Doc-To-Help is finished with preliminary reformatting, exit Word for Windows, restart Word, and open the RTF file.

5. From the **Format** menu, choose **Reformat as Help**.

6. Again, you'll see the **Reformat as Help** dialog box. This time, make sure that only the *second* checkbox is selected, and click **OK**.

7. Repeat steps 4 through 6 for the third, fourth, fifth and sixth stages of the **Reformat as Help** process.

Special Case–Out of Memory During Assign Glossary Definitions Stage

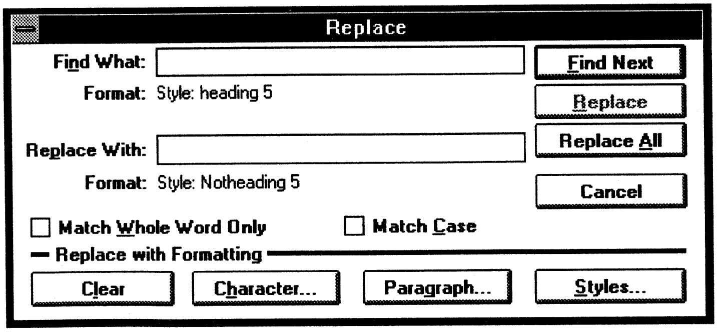
A document with many glossary of terms entries (300 or more) can cause WordBasic out of memory errors during the Assign Glossary Definitions portion of the reformatting process. Unlike the other steps in the Make Into Help process, this step cannot be interrupted and then resumed. A similar problem can occur when a single file project has a modest Glossary of Terms section but many occurrences of these terms within the body of the document. You can usually avoid this problem by making your single-file project into a multifile project. However, in some cases you may not want to convert your single-file project into a multifile project as the single-file project contains cross references that will not work properly in a multifile project.

WexTech Technical Support recommends the following work-around for out of memory errors during the Assign Glossary Definitions process.

1. With the Reformat as Help File dialog box on the screen, turn off the last two steps by r**emoving the check marks** from **Assign Glossary Definitions** and **Final Reformatting**, then click **OK**. Doc-To-Help will perform the first four steps in converting your manual into help.

You can create this style in the RTF template D2H_HELP.DOT before you create the RTF file and it will always be there.

2. Create a style in your RTF file that has the same attributes as Heading 5. For this example, we'll call the style **Notheading5**.
3. Go to your document's glossary of terms section, and scroll down until you are about half way through the section.
4. Place the cursor in front of the first term in the second half of the section (the first term starting with "M" for instance), then select all the terms from this point to the end of the glossary section.
5. From the **Edit** menu choose **Replace**. When the Replace dialog box appears, delete any text in the text edit boxes and use the **Style** button to indicate that you want to replace any instances of Heading 5 style with Notheading5 style, as shown below.



Find What:

Format: Style: heading 5

Replace With:

Format: Style: Notheading 5

□ Match Whole Word Only   □ Match Case

6. Click **Replace All**.

7. Restart Word for Windows, and open the RTF file that contains the glossary of terms. Choose **Format**, **Reformat as Help** and then turn off Final Formatting from the Reformat as Help File dialog box. Click **OK** to run the Assign Glossary Definition process.

8. When the process is completed, go to the glossary of terms section of your document. Select the first half of the terms (the ones that are still in Heading 5 format). Using the techniques described in Step 5, replace all instances of **Heading 5** style with **Notheading 5** style.

9. Select the second half of the terms (the ones previously formatted in Notheading 5 style) and using the techniques described in Step 5, replace all instances of **Notheading 5** style with **Heading 5** style. Save your work and exit Word for Windows.

10. Restart Word for Windows, and open the RTF file that contains the glossary of terms. Choose **Format**, **Reformat as Help** and turn off Final Formatting from the Reformat as Help File dialog box. Click **OK** to run the Assign Glossary Definition process for the second time.

11. When the process is completed, go to the glossary of terms section. Page down to the second half of the terms (or use the Edit Find command to find the first instance of text formatted in Heading 5 style). You will note that the jump text for these terms (single underlined text) has been placed just in front of this first term that you processed in the second pass. Select all these underlined terms, and choose **Cut** from the **Edit** menu. Page up to the first term in the glossary of terms section. You will note it is preceded by the jump text for the first half of the terms. Place your cursor on the empty paragraph mark after the last jump text term. Choose **Paste** from the **Edit** menu to paste the second half of the terms from the Clipboard.

12. Save your RTF file and exit Word for Windows.

13. Restart Word for Windows, and open the RTF file that contains the glossary of terms. Choose **Format**, **Reformat as Help** and click **OK** to run the Final Formatting pass.

These steps can repeated if you have a larger Glossary of Terms and, in theory, there should be no Glossary too large for Doc-To-Help.

Memory Problems When Building the Index

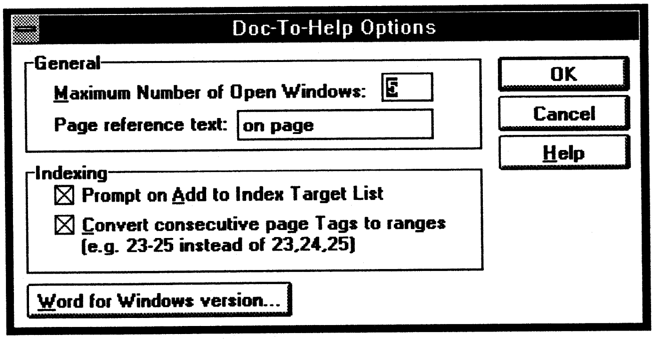
Indexing is another memory-intensive feature which can cause "Out of Memory" errors. First and foremost, if you're getting these errors when trying to build the index, upgrade to Word for Windows 2.0c if you haven't already done so. If you have many inside chapters in a multifile project, or if your index search list is especially long, it may be necessary to use Word 2.0c to complete the index.

Building the index in a multifile Help project requires opening each of the files in the project. By default, Doc-To-Help leaves as many documents open as it can, in order to save time. This may be as many as nine, which is the maximum number of open windows in Word.

If you're working with a multifile project and are encountering "Out of Memory" errors, you can make the Build Index macro less memory-intensive by reducing the number of concurrently open windows. You do this by changing the maximum number of open windows with the **Tools Doc-To-Help Options** command.

To Change the Number of Open Windows

1 From the **Tools** menu choose **Doc-To-Help Options**. The following dialog box will appear.



General

Maximum Number of Open Windows: 5

Page reference text: on page

Indexing

☒ Prompt on Add to Index Target List

☒ Convert consecutive page Tags to ranges (e.g. 23-25 instead of 23,24,25)

2. Reduce the number of open windows in the **Maximum Number of Open Windows** text box. (The default is 5.)

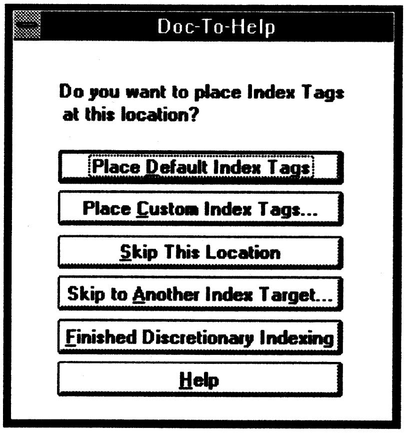
3. Click **OK**.

Memory Errors During Discretionary Indexing

If an Out of Memory error occurs during discretionary indexing, you will probably want to start indexing at the point where the error occurred. This is easily done by using **Skip to Another Index Target**.

To Start Discretionary Indexing After a Memory Error

1. Close and save any open documents.
2. Quit Word for Windows.
3. Restart Word for Windows. Open the document you were indexing (if you were working with a multifile project, open the master document).
4. From the **Tools** menu, choose **Indexing**. When the dialog box appears, choose **Build Index**.
5. Doc-To-Help will find the first occurrence in your document of the first target and display the following dialog box:


Doc-To-Help
Place Default Index Tags
Place Custom Index Tags...
Skip This Location
Skip to Another Index Target...
Finished Discretionary Indexing
Help

Do you want to place Index Tags at this location?

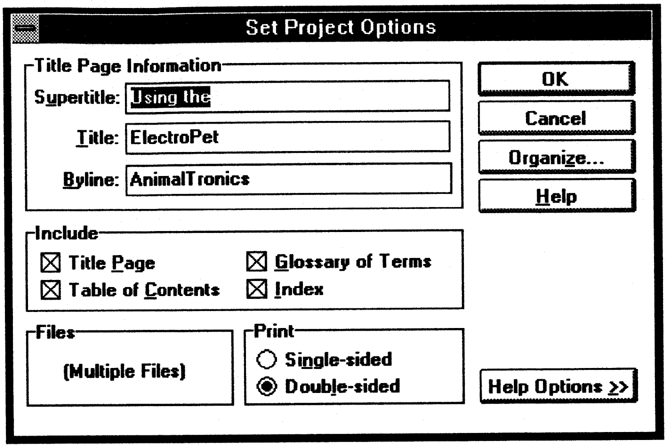
6. Choose **Skip to Another Target**. When the list of targets appears, choose the one you were working with when the Out of Memory error occurred.
7. Continue indexing as usual.

Discretionary Indexing One File at a Time

When you use discretionary indexing to build an index on a multifile project, Doc-To-Help looks for each target through each inside document before searching for the next target. You may find it less of a memory drain to work with one inside file at a time. You can do this by tricking Doc-To-Help into thinking there is only one inside file in the project.

To Index Inside Files Individually

1. Open the master document.
2. From the **Format** menu select **Set Project Options**. The Set Project Options dialog box will appear.



Set Project Options

Title Page Information

Supertitle: Using the

Title: ElectroPet

Byline: AnimalTronics

Include

☒ Title Page ☒ Glossary of Terms

☒ Table of Contents ☒ Index

Files

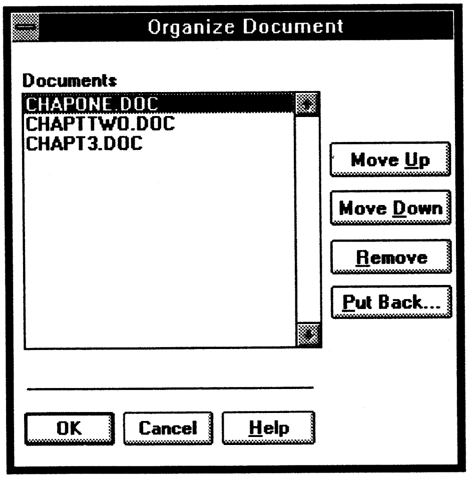
(Multiple Files)

Print

○ Single-sided

● Double-sided

3. Click **Organize.** The Organize Document dialog box will appear, as shown below.



Organize Document

Documents

CHAPONE.DOC
CHAPTTWO.DOC
CHAPT3.DOC

4. Select a file you'll index later. Click **Remove.** You will be temporarily removing all files except the first one you want to index.
5. A message box will appear asking if you want to place the selected document in the REMOVED subdirectory. Click **Yes.**
6. Repeat steps 4 and 5 until you've removed all but one file in the list.
7. Click **OK**. The Set Project Options dialog box will reappear. Click **OK** again.

8. From the **Tools** menu choose **Indexing**. The **Indexing** dialog box will appear.
9. Choose **Build Index**. Build the index as usual (if you want to review the steps for building the index see "Indexing" on page 65).
10. Repeat steps 1 through 3. When the Organize dialog box appears, remove the file you just indexed. Make sure you click **Yes** when you're asked to place the document in the REMOVED subdirectory.
11. Click **Put Back.** A list of files in the REMOVED subdirectory will appear.
12. Select the next file you want to index and click **OK**. When the Organize dialog box reappears click **OK**. When the Document Preferences dialog box reappears click **OK** one last time. Repeat Steps 8 and 9.
13. Repeat steps 1 through 12 until you've built the index for all inside files. When you're finished, you'll need to put back all the documents in the REMOVED subdirectory.

What You Shouldn't Touch and Shouldn't Do

In order for Doc-To-Help commands to work properly, there are certain things that should never be deleted from your manual and/or the Doc-To-Help templates.

Glossaries and Macros

Word for Windows Glossaries are text or graphics you insert from the document template.

Feel free to add your own macros and Word for Windows Glossaries to the Doc-To-Help templates. However, deleting any of the existing macros or glossaries will almost certainly cause problems.

Template Names

Changing the names of the templates will create a problem when converting your manual into Help, or working with a multifile project.

Built-In Style Names

Re-naming or deleting the built-in styles is not recommended.

Structural Changes

The Doc-To-Help macros expect to see a certain structure in your document. If you change the order of the Title Page, Table of Contents, Glossary of Terms or Index sections in the templates or your manual, or delete these sections manually, the results are unpredictable. If you don't want a section in the document, change the settings in **Format Set Project Options.**

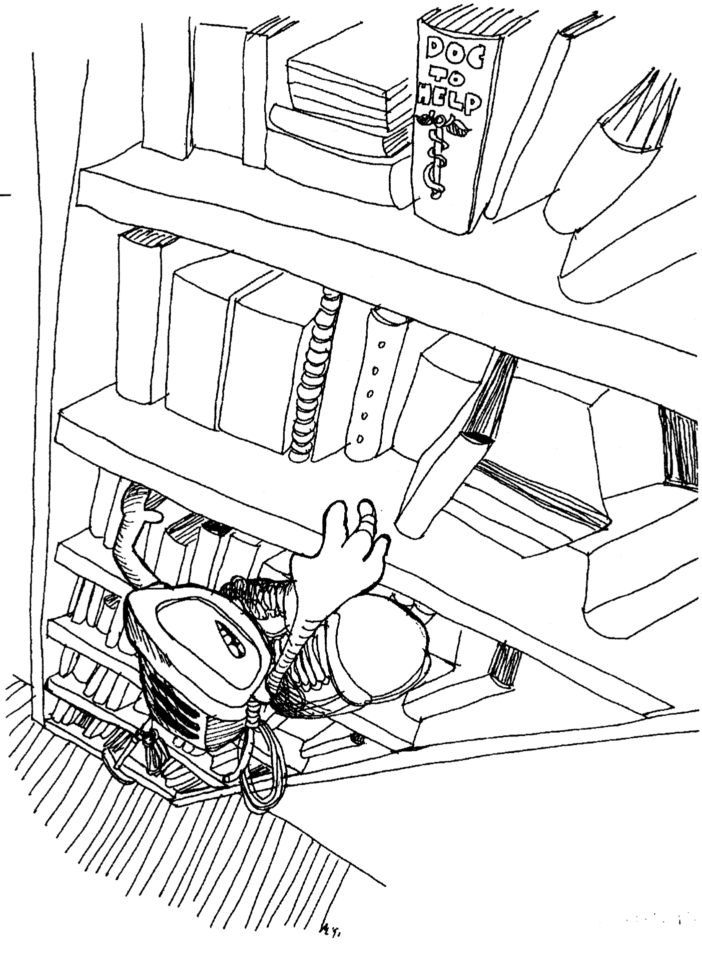
Avoid Renaming Files Using DOS or the File Manager

Wherever possible, try to use either Doc-To-Help commands or Word for Windows' Save As command if you need to rename your files.

Don't use DOS EDIT to edit your HPJ files

The text editor that comes with DOS converts tabs into spaces, which really messes things up if you edit your project's HPJ file and want Doc-To-Help to preserve context string mappings when you update a help file. We suggest using Word for Windows to edit HPJ files.

Reference

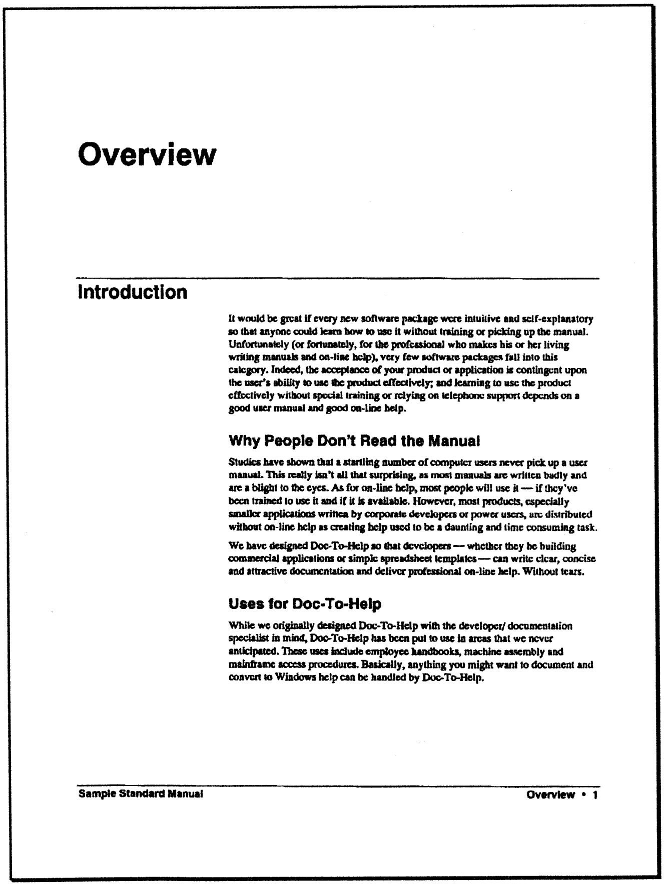


Reference

Doc-To-Help Templates

Standard Manuals (D2H_NORM)

If your manual is divided into separate files, the master document will be based on D2H_NORM and the inside chapters will be based on D2H_IN. See "Adding More Chapters" on page 59.



Overview

Introduction

It would be great if every new software package were intuitive and self-explanatory so that anyone could learn how to use it without training or picking up the manual. Unfortunately (or fortunately, for the professional who makes his or her living writing manuals and on-line help), very few software packages fall into this category. Indeed, the acceptance of your product or application is contingent upon the user's ability to use the product effectively; and learning to use the product effectively without special training or relying on telephone support depends on a good user manual and good on-line help.

Why People Don't Read the Manual

Studies have shown that a startling number of computer users never pick up a user manual. This really isn't all that surprising, as most manuals are written badly and are a blight to the eyes. As for on-line help, most people will use it — if they've been trained to use it and if it is available. However, most products, especially smaller applications written by corporate developers or power users, are distributed without on-line help as creating help used to be a daunting and time consuming task.

We have designed Doc-To-Help so that developers — whether they be building commercial applications or simple spreadsheet templates — can write clear, concise and attractive documentation and deliver professional on-line help. Without tears.

Uses for Doc-To-Help

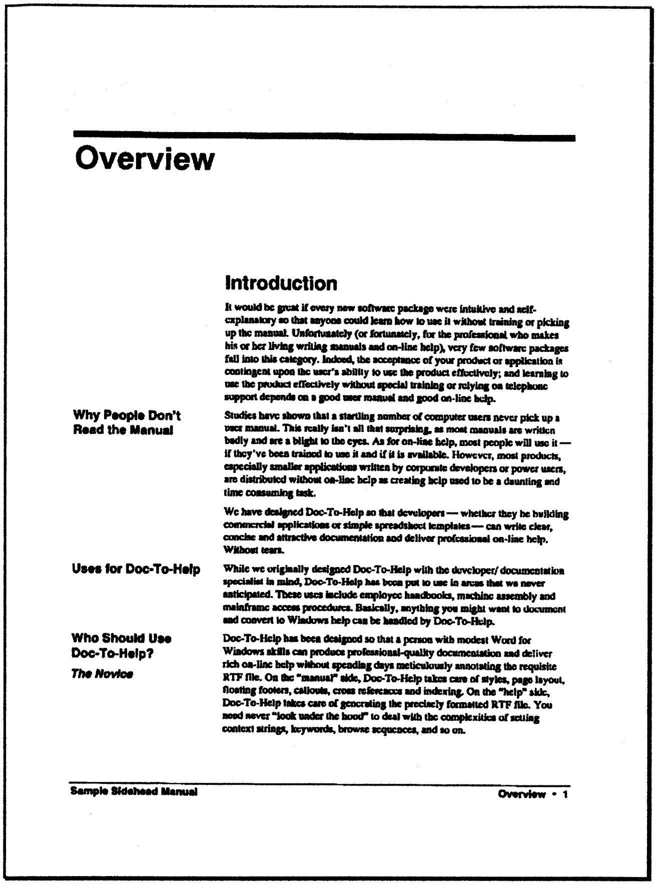
While we originally designed Doc-To-Help with the developer/ documentation specialist in mind, Doc-To-Help has been put to use in areas that we never anticipated. These uses include employee handbooks, machine assembly and mainframe access procedures. Basically, anything you might want to document and convert to Windows help can be handled by Doc-To-Help.

Sample Standard Manual Overview • 1

You create a document that looks like the one shown above by choosing **File New** and choosing **D2H_NORM** from the template list box.

Sidehead Manuals (D2H_SIDE)

If your manual is divided into separate files, the master document will be based on D2H_SIDE and the inside chapters will be based on D2H_INSD. See "Adding More Chapters" on page 59.



Overview

Introduction

It would be great if every new software package were intuitive and self-explanatory so that anyone could learn how to use it without training or picking up the manual. Unfortunately (or fortunately, for the professional who makes his or her living writing manuals and on-line help), very few software packages fall into this category. Indeed, the acceptance of your product or application is contingent upon the user's ability to use the product effectively; and learning to use the product effectively without special training or relying on telephone support depends on a good user manual and good on-line help.

Why People Don't Read the Manual

Studies have shown that a startling number of computer users never pick up a user manual. This really isn't all that surprising, as most manuals are written badly and are a blight to the eyes. As for on-line help, most people will use it — if they've been trained to use it and if it is available. However, most products, especially smaller applications written by corporate developers or power users, are distributed without on-line help as creating help used to be a daunting and time consuming task.

We have designed Doc-To-Help so that developers — whether they be building commercial applications or simple spreadsheet templates — can write clear, concise and attractive documentation and deliver professional on-line help. Without tears.

Uses for Doc-To-Help

While we originally designed Doc-To-Help with the developer/ documentation specialist in mind, Doc-To-Help has been put to use in areas that we never anticipated. These uses include employee handbooks, machine assembly and mainframe access procedures. Basically, anything you might want to document and convert to Windows help can be handled by Doc-To-Help.

Who Should Use Doc-To-Help?

The Novice

Doc-To-Help has been designed so that a person with modest Word for Windows skills can produce professional-quality documentation and deliver rich on-line help without spending days meticulously annotating the requisite RTF file. On the "manual" side, Doc-To-Help takes care of styles, page layout, floating footers, callouts, cross references and indexing. On the "help" side, Doc-To-Help takes care of generating the precisely formatted RTF file. You need never "look under the hood" to deal with the complexities of setting context strings, keywords, browse sequences, and so on.

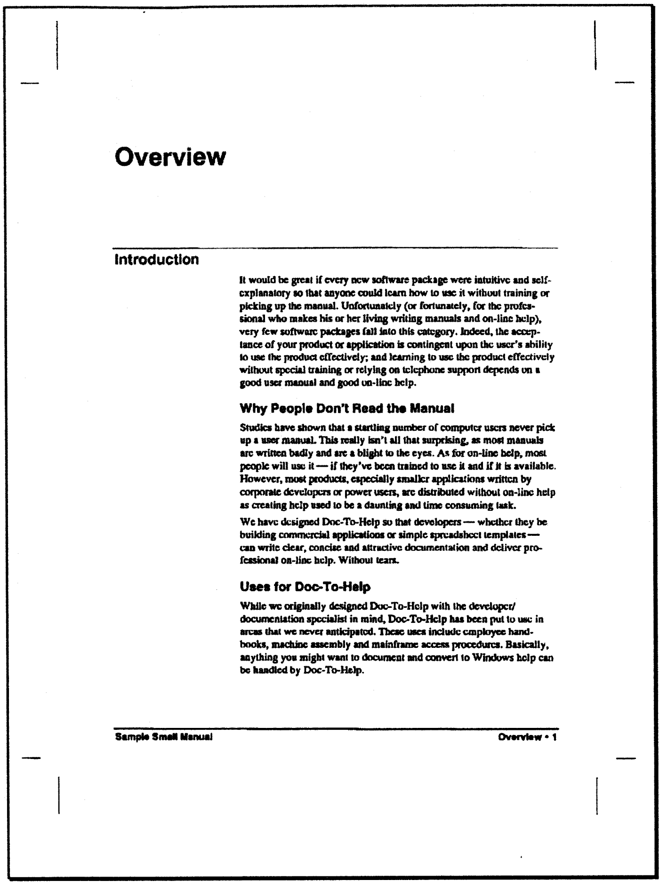
Sample Sidehead Manual   Overview • 1

You create a document that looks like the one shown above by choosing **File New** and selecting **D2H_SIDE** from the template list box.

Small Manuals (D2H_SMAL)

Notice that pages created with this template print out with crop marks.

If your manual is divided into separate files, the master document will be based on D2H_SMAL and the inside chapters will be based on D2H_INSM. See "Adding More Chapters" on page 59.



Overview

Introduction

It would be great if every new software package were intuitive and self-explanatory so that anyone could learn how to use it without training or picking up the manual. Unfortunately (or fortunately, for the professional who makes his or her living writing manuals and on-line help), very few software packages fall into this category. Indeed, the acceptance of your product or application is contingent upon the user's ability to use the product effectively; and learning to use the product effectively without special training or relying on telephone support depends on a good user manual and good on-line help.

Why People Don't Read the Manual

Studies have shown that a startling number of computer users never pick up a user manual. This really isn't all that surprising, as most manuals are written badly and are a blight to the eyes. As for on-line help, most people will use it — if they've been trained to use it and if it is available. However, most products, especially smaller applications written by corporate developers or power users, are distributed without on-line help as creating help used to be a daunting and time consuming task.

We have designed Doc-To-Help so that developers — whether they be building commercial applications or simple spreadsheet templates — can write clear, concise and attractive documentation and deliver professional on-line help. Without tears.

Uses for Doc-To-Help

While we originally designed Doc-To-Help with the developer/documentation specialist in mind, Doc-To-Help has been put to use in areas that we never anticipated. These uses include employee handbooks, machine assembly and mainframe access procedures. Basically, anything you might want to document and convert to Windows help can be handled by Doc-To-Help.

Sample Small Manual Overview • 1

You create a document that looks like the one shown above by choosing **File New** and selecting **D2H_SMAL** from the template list box. Use this template to create documents that will print on 7" by 9" pages.

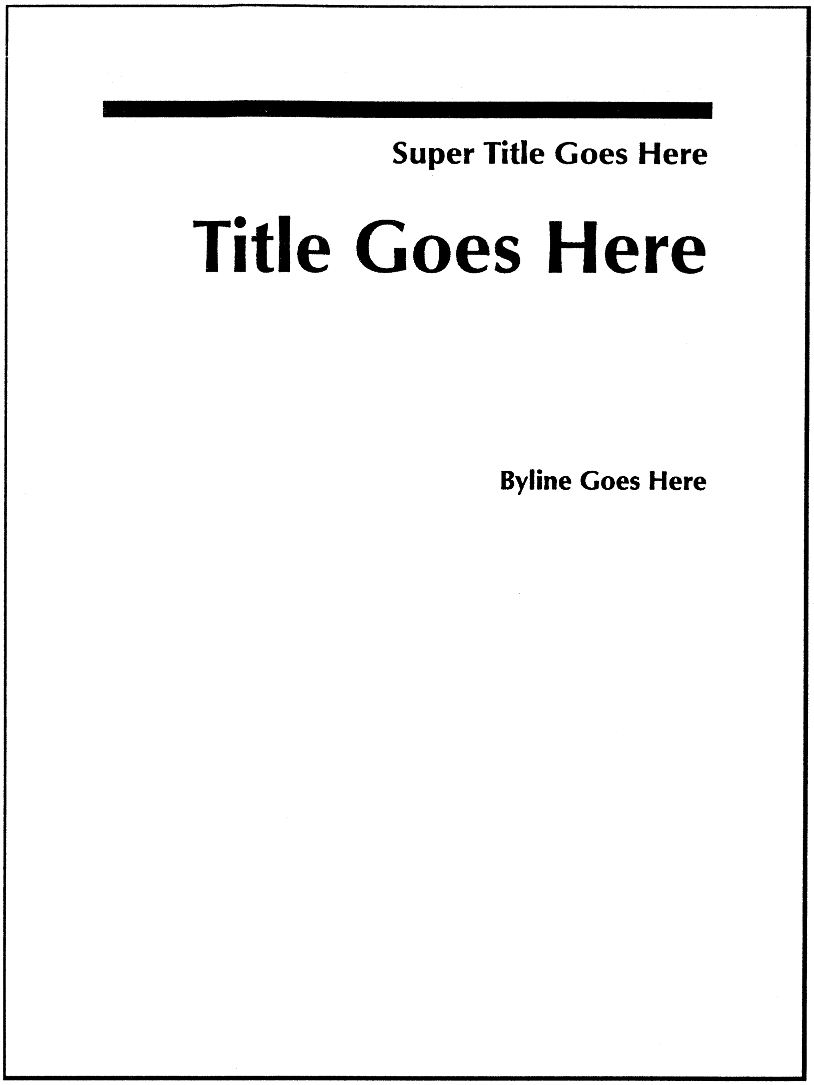
Doc-To-Help Style Definitions

Primary Styles (Styles Often Used)

Style	Shortcut Key	Explanation	Help Equivalent
Body	CTRL+SHIFT+B	Used for body copy.	
BodyTable		Table version of Body.	
Caption		Used for margin callouts and annotations.	Popup definitions.
FigureDesc		Used to annotate full size screen shots.	
Figures		Used to format screen shots.	
FiguresTable		Used to format screen shots contained in tables.	
Heading 1	CTRL+SHIFT+1	Used for the chapter title. This style should be used at the beginning of each chapter.	Depending on the options selected in the Reformat as Help dialog box, Heading 1 will either be static text on the contents page or will be accessible as a "jump" item.
Heading 2	CTRL+SHIFT+2	Used for major section headings.	Will be accessible as a "jump" item from the contents page.
Heading 3	CTRL+SHIFT+3	Used for subheadings.	Will become a related topic of whatever main topic immediately precedes it.
Heading 4	CTRL+SHIFT+4	Used for procedural headings. Step-by-step procedures should be preceded with a title like "To Take Candy From a Baby" formatted in Heading 4 style.	Will become a related topic of whatever main topic immediately precedes it.
List	CTRL+SHIFT+L	Used for bullet and numbered lists.	
ListTable		Table version of List.	
TableHeading		Used to format table headings.	
TableText		Used to format text contained in a table.	

Secondary Styles (Styles Used Less Often)

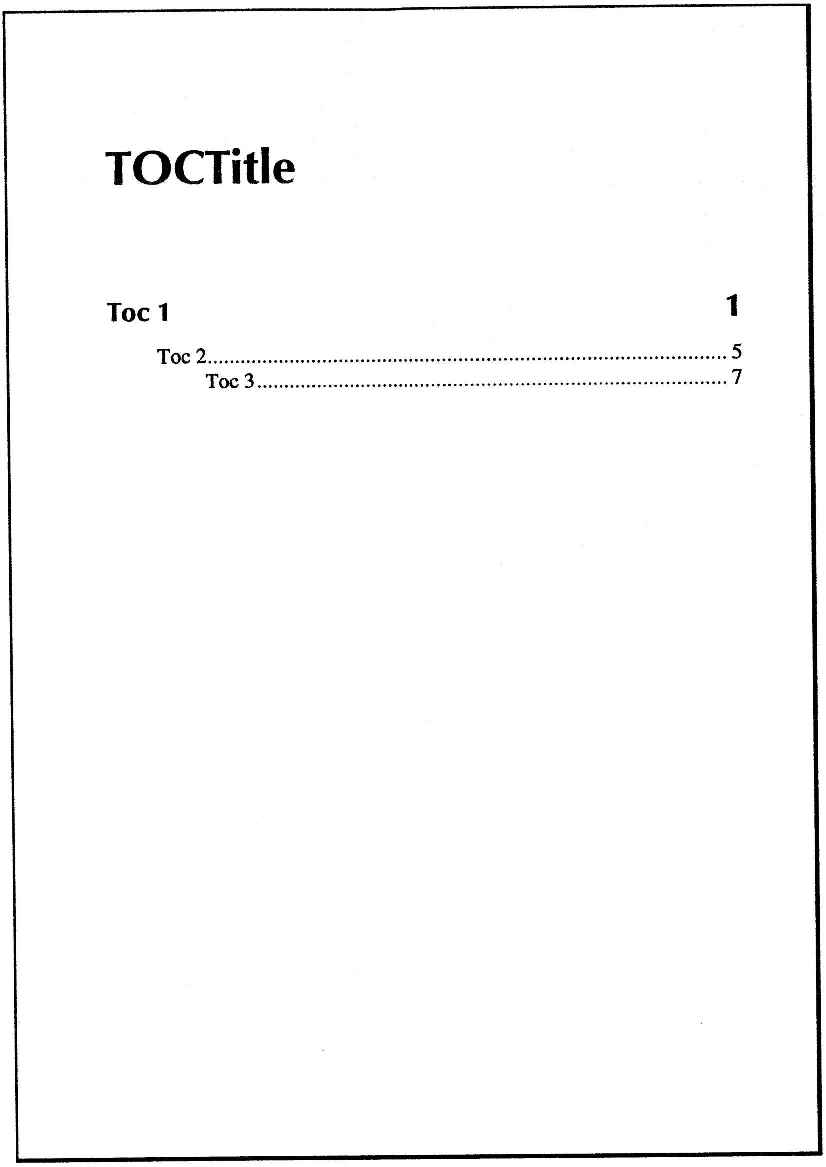
Style	Explanation	Help Equivalent
Byline	Used for title page byline. Applied automatically via Format Set Project Options command.	
CodeExplained	Used to format single lines of source code.	
Definition	Used to format glossary of terms descriptions. Applied automatically via Insert Glossary Term command.	Popup definition.
exHeading3	Used to replace Heading 3 style in the RTF file when text formatted in Heading 3 style is the only heading subordinate to a Heading 2 that has no text immediately following it. This eliminates an unnecessary jump. The lower Heading (topic) becomes part of the topic above it.	
exHeading4	Used to replace Heading 4 style in the RTF file when text formatted in Heading 4 style is the only heading subordinate to a Heading 3 that has no text immediately following it. This eliminates an unnecessary jump. The lower Heading (topic) becomes part of the topic above it.	
Footer	Applied to footers by default.	
Header	Applied to headers by default.	
Heading 5	Used to format glossary of terms headings. Applied automatically via Insert Glossary Term command.	Keyword for popup definition.
index 1	Used to format primary index tags. Applied automatically when index field is updated.	
index 2	Used to format secondary index tags.	
index 3	Used to format tertiary index tags.	
List2	Used for bullet and numbered list items that are subordinate to List style paragraphs.	
List2Table	Table version of List2	
Source	Used to format source code.	
SourceTop	Used to format the first line of source code.	
SuperTitle	Used for title page SuperTitle. Applied automatically via Format Set Project Options command.	
Title	Used for title page Title. Applied automatically via Format Set Project Options command.	
toc 1	Used to format primary table of contents entries. Applied automatically when TOC field is updated.	
toc 2	Used to format secondary table of contents entries. Applied automatically when TOC field is updated.	
toc 3	Used to format tertiary table of contents entries. Applied automatically when TOC field is updated.	
TOCTitle	Used to format the table of contents title on the page that contains the TOC field.	
Warning	Used to format text that needs to grab the reader's attention.	



Super Title Goes Here

Title Goes Here

Byline Goes Here

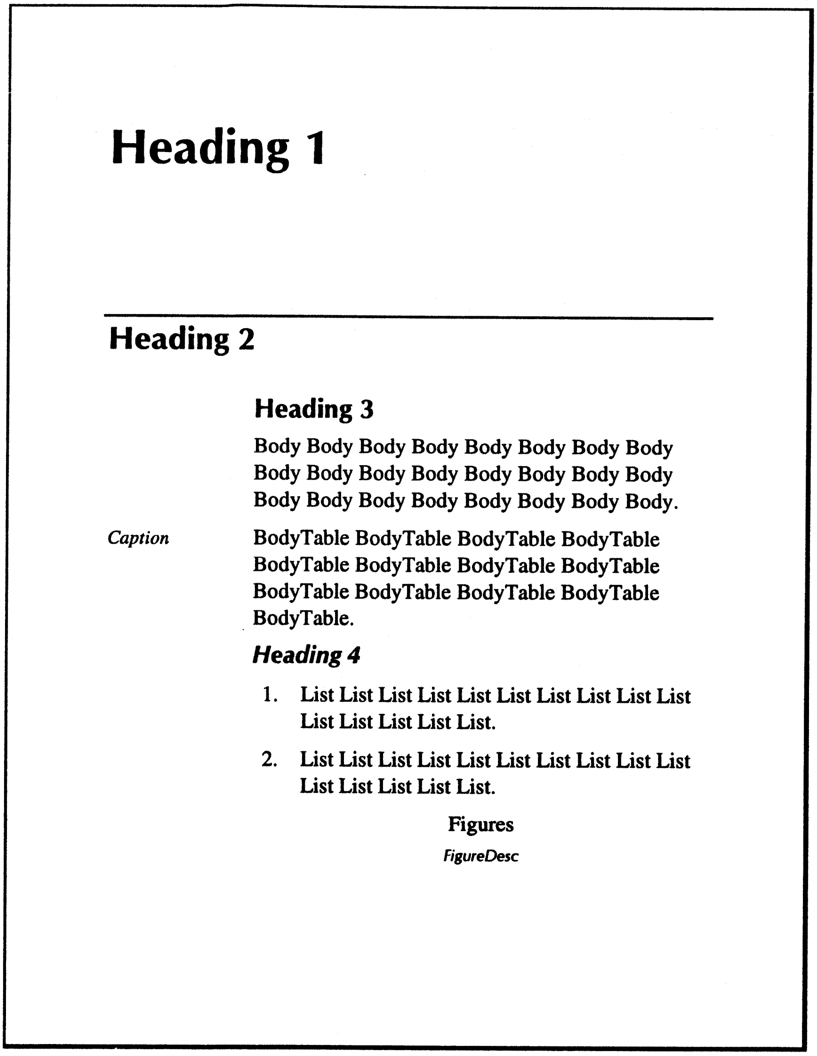


TOCTitle

Toc 1 **1**

Toc 2 5

Toc 3 7



Heading 1

Heading 2

Heading 3

Body Body Body Body Body Body Body Body Body Body Body Body Body Body Body Body Body Body Body Body Body Body Body Body.

Caption

BodyTable BodyTable BodyTable BodyTable BodyTable BodyTable BodyTable BodyTable BodyTable BodyTable BodyTable BodyTable BodyTable.

Heading 4

1. List List List List List List List List List List List List List List List.
2. List List List List List List List List List List List List List List List.

Figures

FigureDesc

***Note:** Heading styles for sidehead template are different.*

Shortcut Keys

Pressing These Keys	Performs This Action
CTRL + NUMERIC KEYPAD PLUS	Adds one point of space after the current paragraph.
CTRL +ALT + NUMERIC KEYPAD PLUS	Adds one point of space before the current paragraph.
CTRL + NUMERIC KEYPAD MINUS	Subtracts one point of space after the current paragraph.
CTRL + ALT + NUMERIC KEYPAD MINUS	Subtracts one point of space before the current paragraph.
CTRL + SHIFT + S	Applies Special Bold formatting.
CTRL + SHIFT +A	Adds selected text to the index list.
CTRL + SHIFT + M	Marks text for manual only.
CTRL + SHIFT + H	Marks text for help only.
CTRL + SHIFT + 1	Applies Heading 1 style.
CTRL + SHIFT + 2	Applies Heading 2 style.
CTRL + SHIFT + 3	Applies Heading 3 style.
CTRL + SHIFT + 4	Applies Heading 4 style.
CTRL + SHIFT + B	Applies Body style. If the insertion point is in a table, applies Body Table style.
CTRL + SHIFT + L	Applies List style. If the insertion point is in a table, applies List Table style.
CTRL + SHIFT + P	Toggles picture place holders on and off.
CTRL + SHIFT + F1	Help on Doc-To-Help.

Doc-To-Help Menu Options

Below is a list of menu items that Doc-To-Help changes and/or adds to Word for Windows' built-in menu items.

Doc-To-Help Manual Templates

File New
File Print

Edit Help Topic Status

View Doc-To-Help Markings

Insert Caption
Insert Link To Caption
Insert Glossary Term
Insert Cross Reference (X-Ref)
Insert Hypertext Link
Insert Help Macro
Insert New Chapter
Insert Coded RTF File

Format Set Project Options
Format Change Fonts
Format Screen Shot
Format Special Bold
Format Adjust Spacing
Format Doc-To-Help Markings
Format Make Into Help

Tools Sort Glossary of Terms
Tools Indexing
Tools Repair Manual
Tools Doc-To-Help Diagnostics
Tools Doc-To-Help Options

Table Standard Table

Help Doc-To-Help
Help Doc-To-Help Keyboard
Help About Doc-To-Help

Doc-To-Help Help Template

Insert Hypertext Jump
Insert Help Macro

Format Reformat as Help

Tools Doc-To-Help Options
Tools Doc-To-Help Setup
Tools Compile
Tools Run Help
Tools Show Context Mapping

File Menu (Manuals–.DOC)

File New

You create a Doc-To-Help manual by choosing **New** from the **File** menu. When the File New dialog box appears, choose the Doc-To-Help template that best meets your needs.

These are the only templates you should access directly. The "inside" templates are accessed automatically when you create a new chapter in a multifile project.

- Use **D2H_NORM** for manuals that will be printed on letter-size or A4 paper.
- Use **D2H_SIDE** for letter-size or A4-size manuals that use side headings.
- Use **D2H_SMAL** for manuals that will be printed on 7" by 9" sheets.

(For samples of these templates, see "Doc-To-Help Templates" on page 195.)

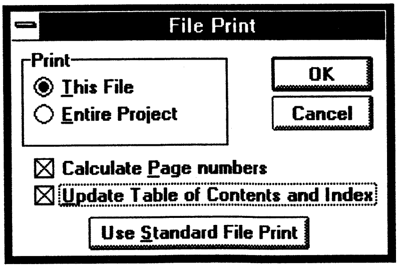
When you create a new document based on one of these templates, the **Set Project Options** dialog box will appear. For more information, see "Format Set Project Options" on page 216.

File Print

Choosing **Print** from the **File** menu will produce different results depending on whether you are working with a single file or multiple files. If you are working with a **single** file, Doc-To-Help will display the standard File Print dialog box. You can choose to update the fields in your document every time you print by pressing the **Options** button in the File Print dialog box and turning off the **Update Fields** check box.

Note: If you update the Table of Contents field before turning off View Help Only text, the Table of Contents in your manual will contain references to text found only in the Help file. To prevent this, make sure you turn off Help-Only text with the ***View Doc-To-Help Markings*** *command before updating the Table of Contents field or printing the document with the Update Fields print option selected.*

If you are working with **multiple** files, Doc-To-Help will display the **Multiple File Print** dialog box, as shown below.



File Print

Print

● This File

○ Entire Project

☒ Calculate Page numbers

☒ Update Table of Contents and Index

Choosing **This File** will print the file you are currently working on while choosing **Entire Project** will print the master document and all files that make up your manual.

Choosing **Calculate Page Numbers** opens each chapter and resets the page numbers so that chapters don't all begin with page one.

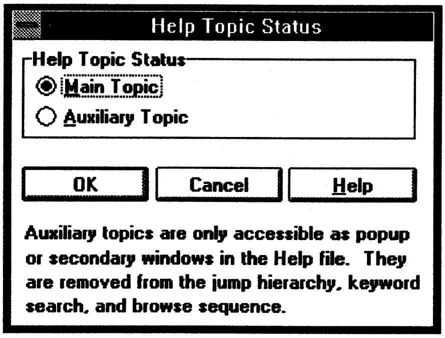
Choosing **Update Table of Contents and Index** will make Doc-To-Help update the TOC and INDEX fields in the master document.

Pressing **Use Standard File Print** will process the options selected in the multifile print dialog box and then display Word's built-in File Print dialog box.

Edit Menu (Manuals–.DOC)

Edit Help Topic Status

Choosing **Help Topic Status** from the **Edit** menu will display the Help Topic Status dialog box, as shown below.



Help Topic Status

● Main Topic

○ Auxiliary Topic

Auxiliary topics are only accessible as popup or secondary windows in the Help file. They are removed from the jump hierarchy, keyword search, and browse sequence.

Choosing **Main Topic** will make the current topic accessible through any of the standard Windows online help access methods (browse sequences, keyword search, etc.).

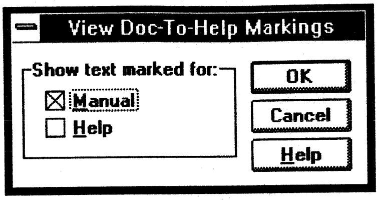
Choosing **Auxiliary Topic** will make this topic accessible only as the result of a link created using the Insert Hypertext Link command.

For a complete discussion of this feature, see "Changing the Help Topic Status" on page 100.

View Menu (Manuals–.DOC)

View Doc-To-Help Markings

Choosing **Doc-To-Help Markings** from the **View** menu will display the View Doc-To-Help Markings dialog box, as shown below.



View Doc-To-Help Markings

Show text marked for:

☒ Manual

□ Help

Text that appears in both the manual and the Help file is displayed in black.

Choosing **Manual** will display text that will appear in your manual (red and black text). Choosing **Help** will display text that will appear in your Help file (magenta and black text). Choosing both will display all manual and help text.

Note: If you choose **View Doc-To-Help Markings** with a multifile manual, Doc-To-Help will ask you if you want to view manual/help markings for the file you're working with or for all files that comprise your manual.

Remember to turn off Help markings when you update the Index field or the Table of Contents fields and before you print the document.

Insert Menu (Manuals–.DOC)

Insert Caption

Use the **Insert Caption** command to place an annotation, comment or graphic in the left margin of the page. You create a caption by

1. Clicking in the paragraph that you want to annotate.
2. Choosing **Caption** from the **Insert** menu. Doc-To-Help will insert a two-column table, with the paragraph you wanted to annotate appearing in the right cell and your cursor positioned in the left cell.

If **Gridlines** is checked in the **Table** menu, you will notice that there are lines outlining the cells of the table. These gridlines do not print out.

Note: These captions will later become popup definitions when you convert the manual into Help. See "Definitions" on page 49.

Insert Link To Caption

Use the **Insert Link To Caption** command to link a caption to text in the manual. When the manual is converted to Help, the caption will become a popup definition of the text that is linked to it.

To Link Text to a Caption

1. Position the insertion point in the caption to which you want to link the text.
2. Choose **Link to Caption** from the **Insert** menu. Doc-To-Help will prompt you to type in the text you want linked to the caption.
3. Enter the text. Doc-To-Help will select the text in the document and ask you to verify that it has selected the correct text.
4. If the correct text has been selected, click **Yes**.

(See "Definitions" on page 49 for more information.)

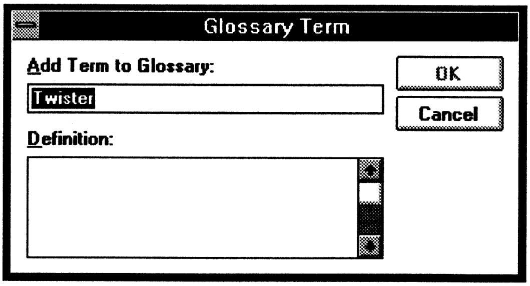
Insert Glossary Term

Note: The limit for glossary terms is 127 characters. The limit for a definition is 4,095 characters.

Use the Insert Glossary Term command to add a new word or phrase to the Glossary of Terms that appears at the end of your document. Make sure that **Glossary of Terms** is selected in the **Format Set Project Options** dialog box before using this command.

To Add a New Term

1. Highlight the term or phrase you want to add.
2. From the **Insert** menu, choose **Glossary Term**. The Glossary Term dialog box will appear.



Glossary Term

Add Term to Glossary: Twister

Definition:

3. You can modify the glossary term in the Add Term to Glossary text box so it looks the way you'd like it to appear in the glossary.
4. Press TAB. Type the definition of the glossary term in the Definition box. To add a hard return in your definitions press CTRL+ENTER.
5. Click **OK**.

These terms are also used to create popup definitions in the Help file. See "Definitions" on page 49.

Note: *If you have lots of glossary entries to add, you can type them in manually in the Glossary of Terms section at the end of your document. Just make sure that the Glossary Term is formatted in the Heading 5 style and the definition is formatted in the Definition style.*

Insert Cross Reference (X-Ref)

Word for Windows has a facility that allows you to enter cross references and page references that change dynamically; that is, if you change the text you want to reference or if you move that text to a different page, Word will update the reference automatically. You can produce these dynamic references by either inserting special field codes or by using Doc-To-Help's **Insert Cross Reference** command.

Before you can use this command, your document must contain **Bookmarks**.

Note: Word for Windows does not support cross references that refer to bookmarks contained in external files. You can only refer to bookmarks contained in the file you are working on.

Also, make sure bookmarks referred to by cross references do NOT include paragraph markings as the cross references may inherit the paragraph properties of the paragraph that contains the bookmark.

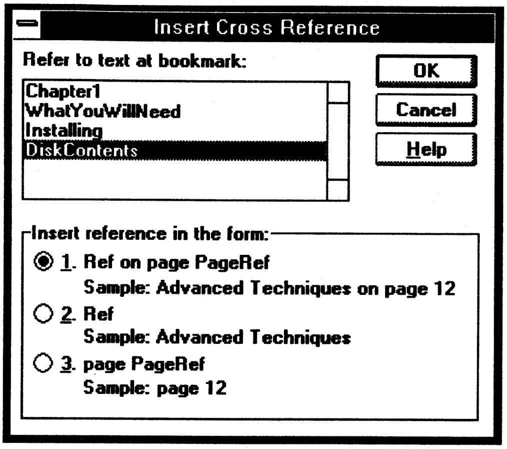
To Insert a Bookmark That Will Be Used as a Cross Reference

1. Highlight the word or phrase you want to reference. Make sure that you do NOT include a paragraph marker.
2. From the **Insert** menu, choose **Bookmark**.
3. When the dialog box appears, enter the name you want to assign to the selected text and click **OK**.

To Insert a Cross Reference/Page Reference

1. Position your cursor where you want the reference to appear.
2. From the **Insert** menu choose **Cross Reference**. The Insert Cross Reference dialog box will appear.

In this example the bookmark "DiskContents" refers to the text "What's on the Disk."



Insert Cross Reference

Refer to text at bookmark:

Chapter1
WhatYouWillNeed
Installing
DiskContents

Insert reference in the form:

● 1. Ref on page PageRef
Sample: Advanced Techniques on page 12

○ 2. Ref
Sample: Advanced Techniques

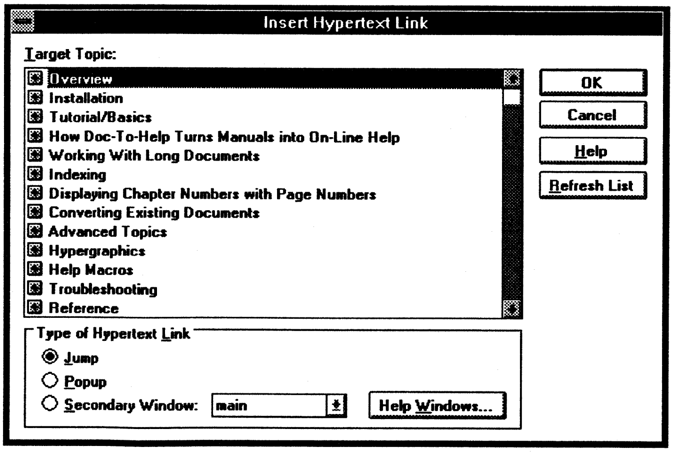
○ 3. page PageRef
Sample: page 12

3. Select the bookmark you want to reference and the type of reference you want to insert.
4. Click **OK**. Doc-To-Help will insert a cross reference/page reference like the one shown below.

For more information, see "What's on the Disk" on page 5.

Insert Hypertext Link

Choosing **Insert Hypertext Link** will display the Insert Hypertext Link dialog box, as shown below.



Insert Hypertext Link

Target Topic:

Overview
Installation
Tutorial/Basics
How Doc-To-Help Turns Manuals into On-Line Help
Working With Long Documents
Indexing
Displaying Chapter Numbers with Page Numbers
Converting Existing Documents
Advanced Topics
Hypergraphics
Help Macros
Troubleshooting
Reference

Type of Hypertext Link

● Jump
○ Popup
○ Secondary Window: main

When you first display this dialog box, the **Target Topic** list box only shows topics formatted in Heading 1 style. To see sub-headings (i.e., subtopics) double-click on any of the topics that have a plus sign (+) next to them. You can also expand a topic using the keyboard by highlighting that topic and pressing the plus sign (+).

Pressing the **Refresh List** button will update the Target Topic list box to reflect any new headings that you have added to your document since the last time you accessed this dialog box.

You specify the type of link you want to create in the **Type of Hypertext Link** group box. The types of links you create are listed below.

Choosing this type of link	Produces this result
Jump	Moves to a different topic within the help file.
Popup	Displays a popup window on top of the current help topic.
Secondary Window	Moves to a different topic within the help file, and displays that topic in a secondary window. (For information on secondary windows, see "Help Windows" on page 223.)

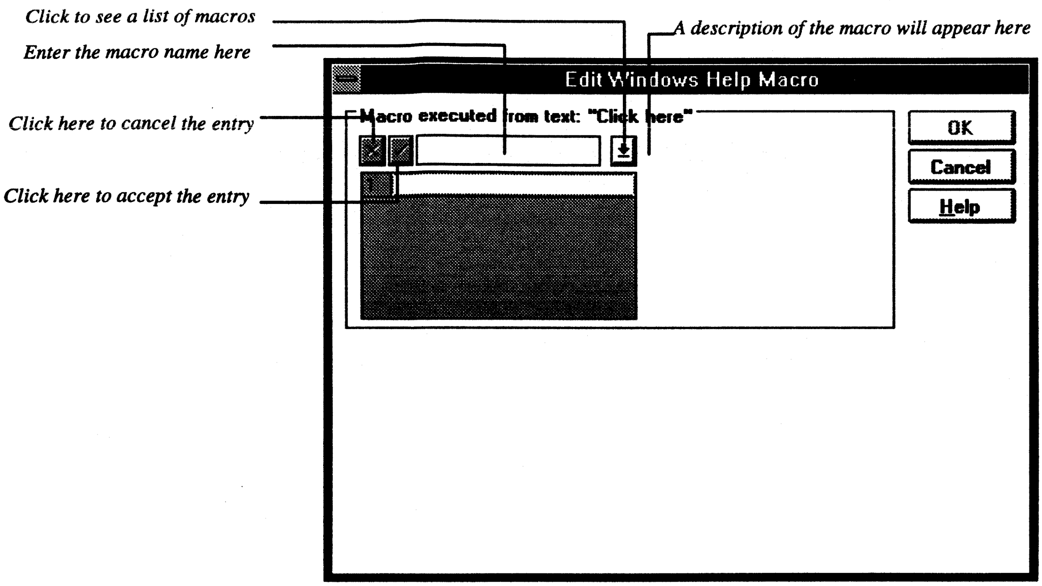
If you choose either popup or secondary window, Doc-To-Help will ask you if you want to designate the target topic as an Auxiliary Help Topic.

If you choose Yes, the only way the users of your help file will be able to access this popup or secondary window jump is through the hypertext link you are now creating. This means that users will not be able to access the topic from the contents topic, a browse sequence or the keyword search facility. You can change how a topic can be accessed using the Edit Help Topic Status For more information, see "Edit Help Topic Status" on page 206.

If you will be using secondary windows, you can access the Doc-To-Help Windows Editor by pressing the **Help Windows** button.

Insert Help Macro

Choosing the **Insert Help Macro** command or clicking **Startup Macros** from within the Set Project Options dialog box will display the Edit Windows Help Macro dialog box, as shown below.


Click to see a list of macros

Enter the macro name here

Click here to cancel the entry

Click here to accept the entry

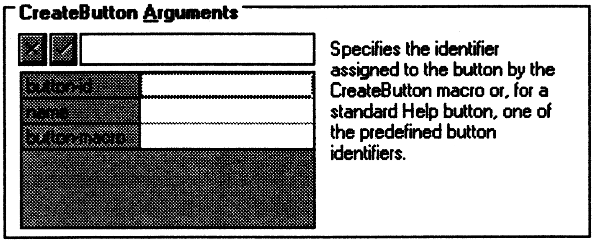
A description of the macro will appear here

Entering macro information in the Doc-To-Help macro editor is a lot like entering information in the cells of a worksheet using a spreadsheet program like Microsoft Excel. Clicking on the check icon will finalize your entry while clicking the X will cancel your entry.

Filling in the Blanks

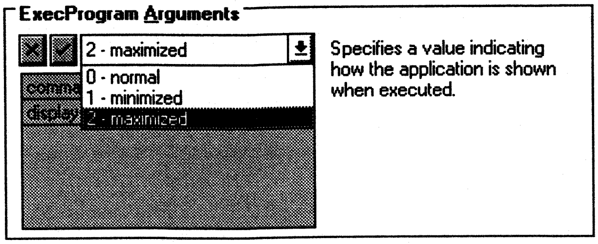
Most macros take at least one argument. When you enter a macro that requires an argument, the **Arguments** group box will appear, as shown in the CreateButton example below.

The CreateButton macro takes three arguments. Clicking next to the argument name changes the description that appears.



Specifies the identifier assigned to the button by the CreateButton macro or, for a standard Help button, one of the predefined button identifiers.

The dialog box will assist you whenever possible by providing descriptions for the arguments and/or by giving you list of available arguments from which to choose. An example of a drop-down list that appears when editing the ExecProgram macro is shown below.

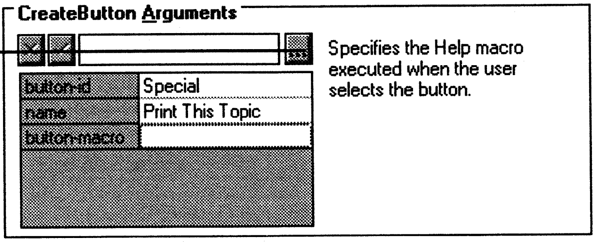


Specifies a value indicating how the application is shown when executed.

Some macros actually take other macros as arguments. For example, when using the CreateButton macro you need to specify the name and ID of the buttons as well as enter the help macro you want run when the button is pressed. Whenever you edit an argument that requires the editing of an additional macro a special ellipsis button will appear, as shown below.

Clicking here will spawn a second Edit Windows Help macro dialog box.

You can access this button with the keyboard by holding down the Alt key and pressing period (.).



button-id	Special
name	Print This Topic
button-macro	

Specifies the Help macro executed when the user selects the button.

Clicking the ellipsis button will spawn a second Edit Windows Help Macro dialog box on top of the first.

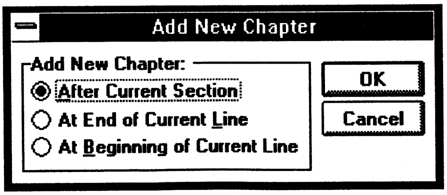
For more information, see "Help Macros" on page 149.

Insert New Chapter

Use the **Insert New Chapter** command to add new chapters to your manual. This command will behave differently depending on whether you have indicated you are using a single file or multiple files in the Document Preferences dialog box.

Single File

If you are creating your manual using a **Single** file, the Add New Chapter dialog box will appear, as shown below.



Add New Chapter:

● After Current Section

○ At End of Current Line

○ At Beginning of Current Line

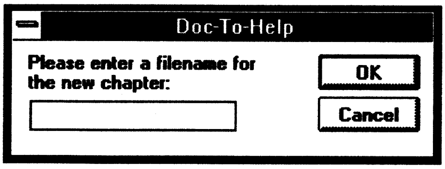
Choosing **After Current Section** will make Doc-To-Help go to the end of the current chapter and insert a section break. This new section will inherit the properties of the previous section (odd and even pages, page numbering scheme, and so on).

Choosing **At End of Current Line** will make Doc-To-Help go to the end of the current line and insert a section break. This new section will inherit the properties of the previous section.

Choosing **At Beginning of Current Line** will make Doc-To-Help go to the beginning of the current line and insert a section break. This new section will inherit the properties of the previous section.

Multiple Files

If you are creating your manual using the **Multiple** file setting, the New Chapter dialog box will appear, as shown below.



Doc-To-Help

Please enter a filename for the new chapter:

New chapters will be based on a different template from master documents. If your master document is based on D2H_NORM (for standard manuals), the inside chapters will be based on D2H_IN. If the master document is based on D2H_SIDE (for sidehead manuals), the inside chapters will be based on D2H_INSD, and if the master document is based on D2H_SMAL (for manuals that will be printed on smaller paper), the inside chapters will be based on D2H_INSM.

Insert Coded RTF File

See "Working with Pre-Existing RTF Files" on page 93.

Insert Menu (Help–.RTF)

Insert Hypertext Jump

The **Insert Hypertext Jump** command allows you to place hypertext jumps in your RTF file, the file Doc-To-Help creates from your manual. Basically, this is an "I want to do something at the last minute" version of the Insert Hypertext Link command available when you are editing your manual. We use the term "last minute" because anything you add to the RTF file will not appear in the manual, so if you change the manual and then convert the changed manual into Help, the hypertext jump you placed in the RTF file will be lost.

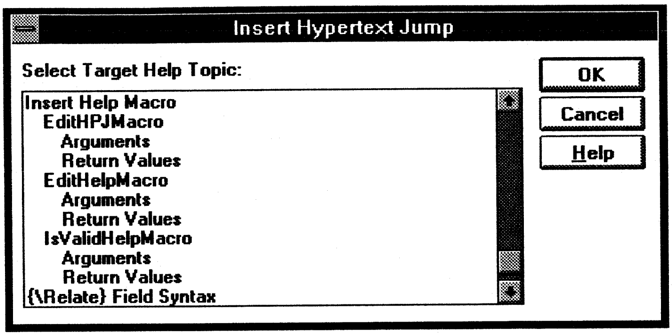
You can create a jump based on a word or phrase in the topic screen or you can create a list of jumps that are not associated with a word or phrase (you can use this method to add to the related topics list).

Remember that converting the document into Help will overwrite the RTF file so related topics you insert in the RTF will be lost the next time you make the document into Help. If, however, you know you won't need to change the manual, or if you are in a hurry and don't want to wait for

Doc-To-Help to convert the manual all over again, you can do the following:

To Create a Jump Based on A Word, Phrase or Graphic

1. Highlight the word, phrase or graphic you want to associate with the hypertext jump. (In the Help file, clicking on the highlighted text will jump to the topic selected.)
2. Choose **Hypertext Jump** from the **Insert** menu. The Insert Hypertext Jump dialog box will appear, as shown below.



Insert Help Macro
EditHPJMacro
Arguments
Return Values
EditHelpMacro
Arguments
Return Values
IsValidHelpMacro
Arguments
Return Values
{\Relate} Field Syntax

3. Choose the topic you want to jump to and click **OK**.

To Create a List of Jumps

1. Position the insertion point in front of an empty paragraph mark in the topic where you want the list to appear.
2. From the **Insert** menu, choose **Hypertext Link**. The Insert Hypertext Jump dialog box will appear.
3. Choose the first topic you want in the list and click **OK**.
4. The dialog box will reappear. Select the next topic for the list. Click **OK**.
5. Continue inserting topics until you've built the list. Press **Cancel** when you're finished.
6. A list of jumps will be created in the RTF file. You'll notice that the text of the headings you selected will appear as the button text for each jump.

Insert Help Macro

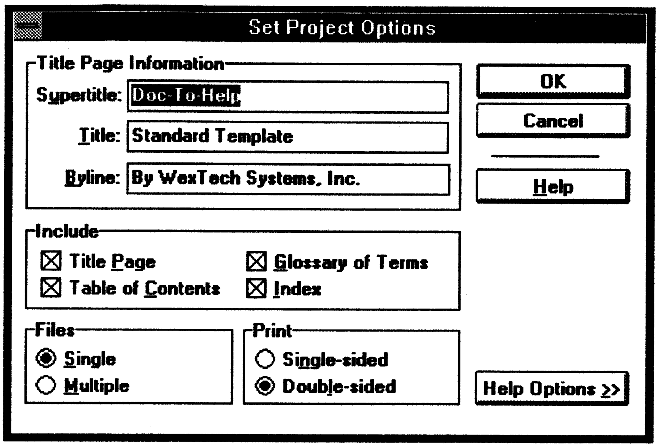
The **Insert Help Macro** command allows you to place help macros in your RTF file. Basically, this is a "last minute" version of the same command available when you are editing your manual. We use the term last minute because anything you add to the RTF file will not appear in the manual, so if you change the manual and then convert the changed manual into Help, the macro you placed in the RTF file will be lost.

See "Insert Help Macro" (manuals) on page 211.

Format Menu (Manuals-.DOC)

Format Set Project Options

Choosing the **Format Set Project Options** command will display the Document Preferences dialog box, as shown below.



Title Page Information

Supertitle: Doc-To-Help

Title: Standard Template

Byline: By WexTech Systems, Inc.

Include

☒ Title Page ☒ Glossary of Terms

☒ Table of Contents ☒ Index

Files

● Single

○ Multiple

Print

○ Single-sided

● Double-sided

Your document's footer will contain whatever has been placed in the Title text box of the File Summary Info dialog box. If you want the footer to be different from the text on the cover page, make the change manually in File Summary Info.

Through this dialog box you can edit the **Supertitle** (which appears on the cover page, above the title), the **Title** and the **Byline**. When you save your file the first time, the Supertitle and Title information are combined and placed in the Title section of your document's File Summary Info dialog box.

You can also determine whether you want a **Table of Contents** section, an **Index**, **Glossary of Terms** or a **Title Page** placed in the document. By default, these check box items are turned on.

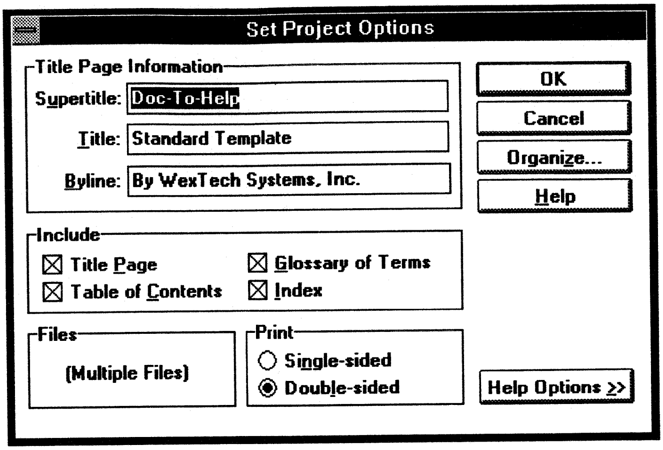
You should also use this dialog box to determine whether or not you will be printing on one side of the page (**single-sided**) or both sides (**double-sided**) where double-sided means that the layout will be different for odd and even pages. With double-sided selected, page numbers will appear in the outside margin and the page setup is modified so that there is more space allotted to the inside margin.

If your manual will be long (over 200 pages), you'll probably want to choose **Multiple** instead of **Single** in the Files group box. Choosing Multiple indicates that you want to break up your manual into a master file and several supporting files. For more information, see "Working With Long Documents" on page 57.

You can access help-specific settings by clicking the Help Options Button.

Multifile Projects and the Organize Command

Choosing this command with a multifile project will present a slightly different version of the single-file Document Preferences dialog box, as shown below.



Title Page Information

Supertitle: Doc-To-Help

Title: Standard Template

Byline: By WexTech Systems, Inc.

Include

☒ Title Page ☒ Glossary of Terms
☒ Table of Contents ☒ Index

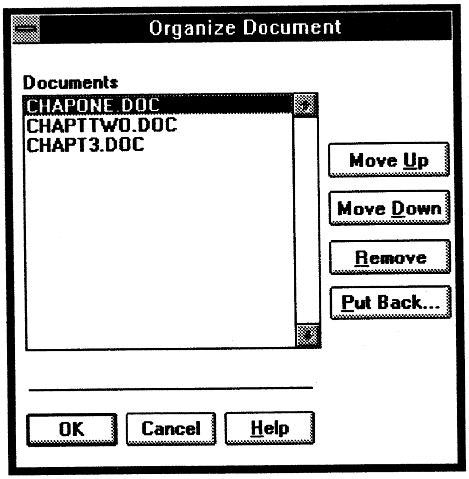
Files

(Multiple Files)

Print

○ Single-sided
● Double-sided

Pressing **Organize** will display the Organize Document dialog box, as shown below.



Documents

CHAPONE.DOC
CHAPTTWO.DOC
CHAPT3.DOC

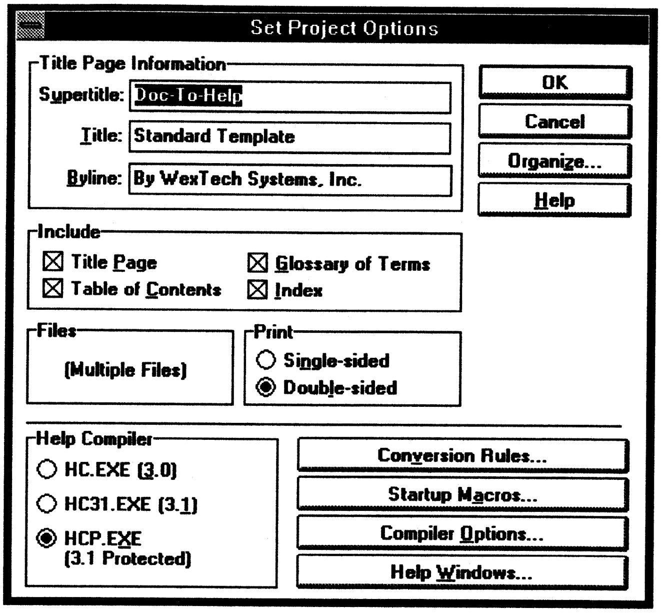
The Organize Document dialog box displays the order in which all the inside chapters of your manual will be printed. Click **Move Up** to move a chapter up in the list and **Move Down** to move a chapter down in the list. Click **Remove** to remove a file. Click **Put Back** to put it back. Removing a document only removes the reference to it in the master document, it does *not* delete it from the disk. You will be prompted to store the document in a directory called REMOVED so it can be **Put Back** into the document later.

Note: You can incorporate an existing chapter into a help project by copying it to the REMOVED directory and then bringing it back with **Put**

Back. Doc-To-Help will verify that the file is based on the correct inside template. (For more information on multifile projects, see "Working With Long Documents" on page 57.)

Format Set Project Options—Help Options

Clicking the **Help Options** button expands the Set Project Options dialog box as shown below.



Set Project Options

Title Page Information

Supertitle: Doc-To-Help

Title: Standard Template

Byline: By WexTech Systems, Inc.

Include

☒ Title Page ☒ Glossary of Terms

☒ Table of Contents ☒ Index

Files

(Multiple Files)

Print

○ Single-sided

● Double-sided

Help Compiler

○ HC.EXE (3.0)

○ HC31.EXE (3.1)

● HCP.EXE (3.1 Protected)

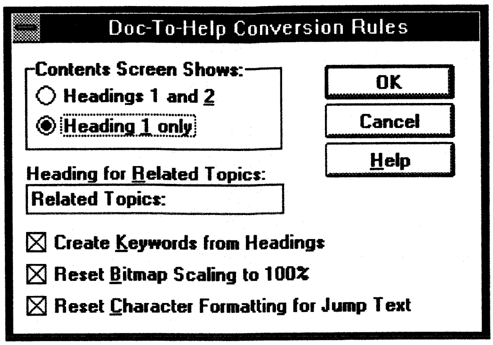
Help Compiler

Doc-To-Help installs three different versions of the Microsoft Help Compiler. HC.EXE will create help files that are compatible with the Windows 3.0 help engine (WINHELP.EXE). HC31.EXE and HCP.EXE create help files that are compatible with the Windows 3.1 help engine. HCP.EXE is actually just a special version of HC31.EXE that takes advantage of extended memory. If you will be compiling large help files, or help files with many graphics, you'll probably need to use HCP.EXE.

The Windows 3.1 help engine supports tables, help macros, scalable fonts, non-scrolling regions and sports an improved user interface. Even if you are using Windows 3.0, you can still work with Windows 3.1 help files as long as you use the Windows 3.1 help engine. Doc-To-Help installs the 3.1 version of WINHELP.EXE in your Windows directory if you are using Windows 3.0.

Conversion Rules

Clicking Conversion Rules will display the Conversion Rules dialog box, as shown below.



Doc-To-Help Conversion Rules

Contents Screen Shows:
○ Headings 1 and 2
● Heading 1 only

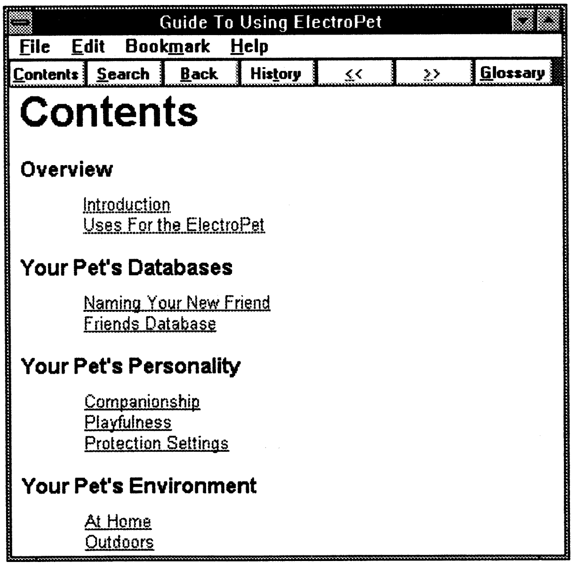
Heading for Related Topics:
Related Topics:

☒ Create Keywords from Headings
☒ Reset Bitmap Scaling to 100%
☒ Reset Character Formatting for Jump Text

Contents Screen Shows

The contents screen is based on your document's table of contents (if you don't have a table of contents, Doc-To-Help will create one for you). You decide how much detail you want to show on the contents page using the **Contents Screen Shows** option buttons. If your manual is short, you'll probably want to use the **Heading 1 and 2** option which will show all text formatted in Heading 1 and 2 styles, as shown below.

Text formatted in Heading 1 style will appear as static text while text formatted in Heading 2 style will become cross references (jumps) to other topics.



Contents

Overview

Introduction
Uses For the ElectroPet

Your Pet's Databases

Naming Your New Friend
Friends Database

Your Pet's Personality

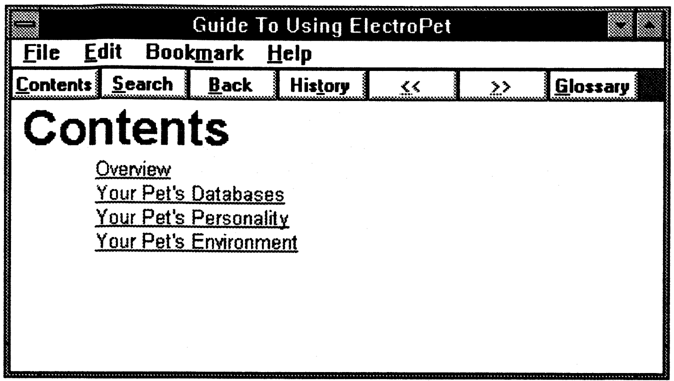
Companionship
Playfulness
Protection Settings

Your Pet's Environment

At Home
Outdoors

Sample Help showing Headings 1 and 2 on Contents Page

Choosing **Heading 1 only** will produce a less detailed contents page, showing only text formatted in Heading 1 style.


Guide To Using ElectroPet
File Edit Bookmark Help
Contents Search Back History << >> Glossary

Contents

Overview
Your Pet's Databases
Your Pet's Personality
Your Pet's Environment

Sample Help showing Heading 1 only on Contents Page

Heading for Related Topics

Anything formatted in Heading 2, Heading 3 or Heading 4 style becomes a related topic of whatever main topic immediately precedes it. You can change the text that appears above this related topic by changing what appears in the **Heading for Related Topics** edit box. Possible entries include, “Sub-Topics,” “See Also” and “More Details.”

Create Keywords from Headings

Turning this option on will make Doc-To-Help convert all your headings into keyword entries that will populate the keyword search dialog box. This option does not affect the conversion of index fields (XE fields) into help keywords.

Reset Bitmap Scaling to 100%

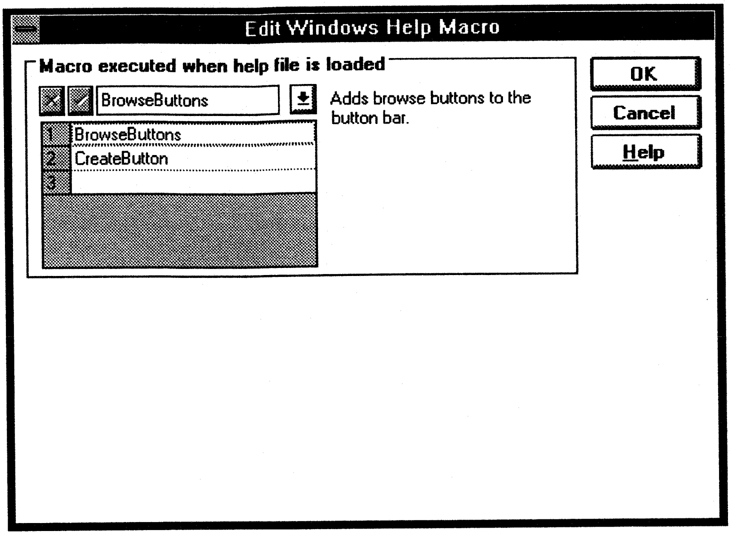
Turning this option on will reset the scaling of bitmaps to 100% in the help file. This option is set to on by default as most scaled bitmaps (which look fine when you print) do not look good on the screen.

Reset Character Formatting for Jump Text

Turning this option on will reset character formatting on text that will become hypertext jumps to that of the current paragraph’s style during **Make Into Help**. This is used most often on glossary terms and text linked to captions that are marked as **Special Bold** in the manual. This will make all jump text in the Help screens appear in a consistent character format regardless of any special character formatting that was applied to the term in the manual (i.e. italics or Special Bold).

Help Macros

Clicking Help Macros will display the Edit Windows Help Macros dialog box (an example is shown below).



Macro executed when help file is loaded

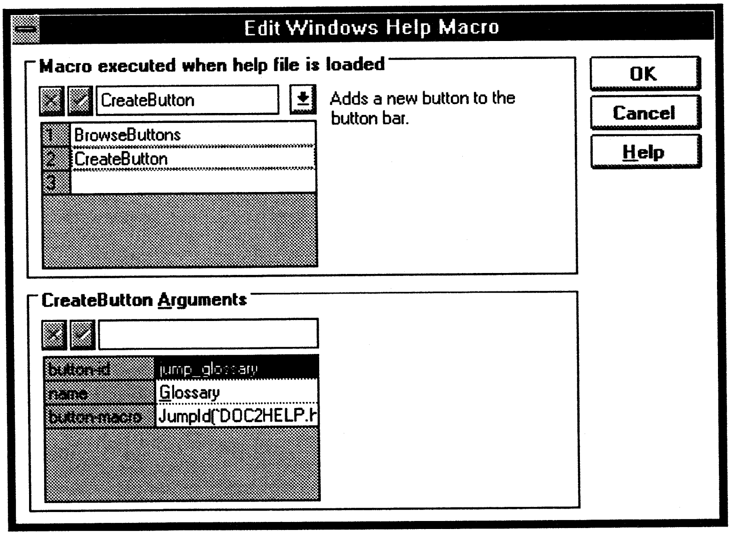
BrowseButtons

Adds browse buttons to the button bar.

1 BrowseButtons
2 CreateButton
3

Doc-To-Help will not add the CreateButton macro if you do not have a glossary of terms in your document.

By default, Doc-To-Help adds the BrowseButtons and CreateButton macros to the list of Startup macros stored in the [Config] section of the Help project file (HPJ file). If you click on the CreateButton macro, The arguments edit section will appear, as shown below.



Macro executed when help file is loaded

CreateButton

Adds a new button to the button bar.

1 BrowseButtons
2 CreateButton
3

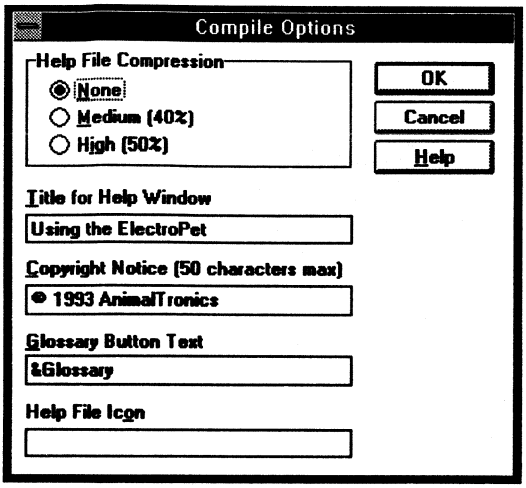
CreateButton Arguments

button-id	jump_glossary
name	Glossary
button-macro	JumpId(`DOC2HELP.H

You can add additional macros by clicking in the first empty cell. For information on adding and deleting macros, see "Insert Help Macro" on page 211 and "Help Macros" on page 149.

Compiler Options

Clicking Compiler Options will display the Compile Options dialog box (an example is shown below).



Compile Options

Help File Compression

● None

○ Medium (40%)

○ High (50%)

Title for Help Window: Using the ElectroPet

Copyright Notice (50 characters max): © 1993 AnimalTronics

Glossary Button Text: &Glossary

Help File Icon:

Help File Compression

The Windows 3.1 Help compilers (HC31.EXE and HCP.EXE) offer three compression options. Choosing **None** will compile fastest, but will produce a large HLP file. Choosing **Medium** will compile less quickly, but will produce a smaller HLP file. Choosing **High** will result in the slowest compile time, but the smallest HLP file.

Title

By default, the Title of your document will appear as the caption for the Help window. You can change this title as needed.

Glossary Button Text

If you are compiling for Windows 3.1 help, Doc-To-Help will add an additional button to the six standard buttons that appear below the Help window menu bar. Clicking this button will jump to your help file's glossary of terms. By default, the text for this button is **Glossary**.

You can create your own buttons using Help macros. For more information, see "Help Macros" on page 149.

Help File Icon

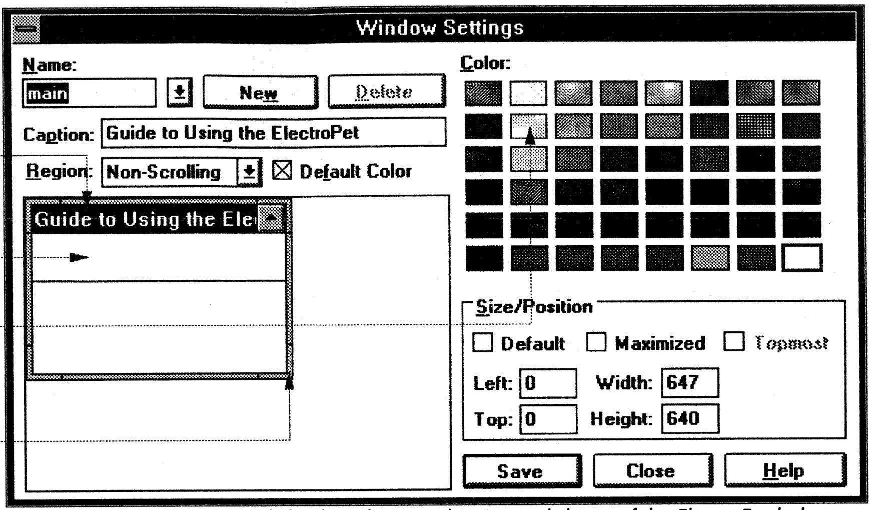
There are a variety of icon editing utilities available, including Iconworks which comes with Visual Basic.

By default, when you minimize your help file the standard question mark icon will appear. If you're a little tired of this plain yellow icon, you can create your own icon or use shareware icons.

You specify the custom icon you want to use by placing the name of the icon in the Help File Icon edit box. If the icon file is not in the same directory as your Doc-To-Help project, you will need to specify the drive and path name as well as the filename.

Help Windows

Clicking Help Windows will display the Windows Settings dialog box (an example is shown below).



Window Settings

Name: main

Caption: Guide to Using the ElectroPet

Region: Non-Scrolling

☒ Default Color

Size/Position

□ Default □ Maximized □ Topmost

Left: 0 Width: 647

Top: 0 Height: 640

Click on the title and drag to move the help window.

Click here to select the non-scrolling region.

Click here to select a color for the selected region.

Click here and drag the border to change the size of the window.

The Window Settings dialog box showing the size and shape of the ElectroPet help window relative to a full-sized screen.

A secondary window is an independent Help window that has a title bar and scroll bars but no menu bar or buttons. Secondary windows are resizable and moveable, and they remain open until closed by the user or by a help macro, whether or not the main Help window is open.

The Windows Settings dialog box allows you to control the size, placement and colors of the main help window and up to five secondary windows (five being the Microsoft Help Compiler limit).

When you first create a project with Doc-To-Help, there is one window called main with a caption that is the same as your document title. You can change the caption, if you want.

You access the window whose settings you want to change through the **Name** drop down list box.

You create a new window by clicking the **New** button, and you delete a window by clicking the **Delete** button. You cannot delete the main window. When you create a new window, Doc-To-Help will name the window "new" and assign it a caption of "new." You can change the window name and the caption as desired. The window name and associated caption do not have to be the same.

You can control the **colors** of the windows' non-scrolling and scrolling regions by selecting the region you want to change, and then choosing the desired color. You can access these regions either by clicking on the region in the sample window, or by choosing from the **Region** drop-down list box. Turning the **Default Color** option on will change the selected region's color to white.

You can change the size and placement of the active window directly manipulating the sample window or by specifying coordinates in the **Size/Position** group box. The Microsoft Help Compiler uses a relative coordinate system, so the window size you specify will take up the same amount of screen space no matter what resolution display is being used. Turning the **Default** option on will cause the window to be displayed using the default settings found in the [Windows Help] section of the user's WIN.INI file; the size/position settings specified in this dialog box will be ignored. Turning the **Maximized** option on will make the window take up the full screen when the window is first displayed. Turning the **Topmost** setting on will make a secondary window stay on top of other windows. The Topmost option is only available when you are working with a secondary window.

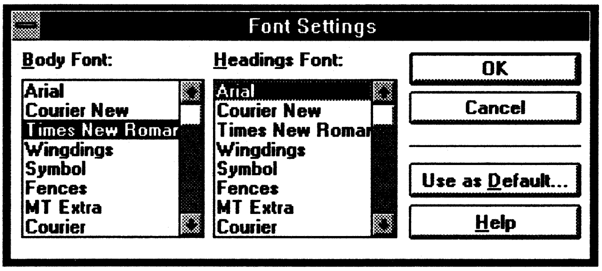
Clicking **Save** will write the windows settings to the Help Project file (HPJ file.) Clicking **Close** will close the Windows Settings dialog box.

Format Change Fonts

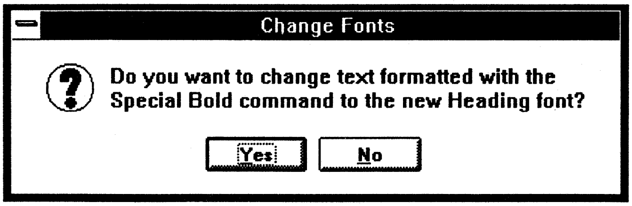
If you are using Windows 3.0, the list of fonts displayed will be different.

By default, Doc-To-Help templates have been formatted in True Type fonts. Headings are in Arial bold and body text is in Times New Roman. You can change these settings using the **Format Change Fonts** command.

Choosing the command will display the Font Settings dialog box, as shown below.



Select the fonts you want to use and click **OK**. Clicking **Use as Default** will alter the underlying template so that all new documents based on this template will use the fonts that you choose. If you change the heading font, the Change Fonts dialog box will appear, as shown below.



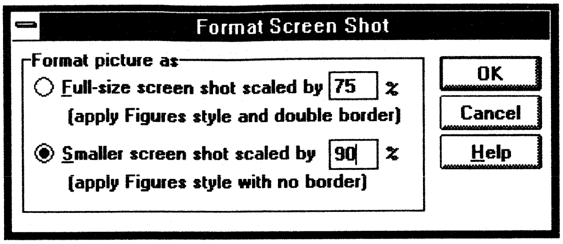
Do you want to change text formatted with the Special Bold command to the new Heading font?

Note: This command will change the fonts in your manual but will not affect the fonts in the Help file. For information on how to change the fonts in the Help file see "Redefining Styles in Help" on page 107.

When this dialog box appears, Doc-To-Help is asking you if you want to search for any text directly formatted using the **Special Bold** command. For example, let's suppose that half-way through writing a manual you decide to change your heading font from Helvetica to Avant Garde. If you had previously used the Special Bold command to format text in Helvetica bold, you will probably want Doc-To-Help to find that text and replace it with Avant Garde bold, so click **Yes**.

Format Screen Shot

Use the **Format Screen Shot** command to scale and format screen shots. Choosing this command will display the Format Screen Shot dialog box, as shown below.



Format picture as

○ Full-size screen shot scaled by 75 %
(apply Figures style and double border)

● Smaller screen shot scaled by 90 %
(apply Figures style with no border)

Selecting **Full-size screen shot** will scale the graphic, apply Figures style formatting and place a double border around the graphic. Choosing **Smaller screen shot** (which you should use to format screen shots of dialog boxes and smaller screen components) will scale the graphic and apply Figures style formatting.

The following scaling factors work well with Doc-To-Help templates and screen shots produced in standard VGA mode.

Template	Screen Shot Type	Scaling Factor
D2H_NORM	Full	70%
D2H_NORM	Small (for dialog boxes, etc.)	90%
D2H_SIDE	Full	70%
D2H_SIDE	Small	90%
D2H_SMAL	Full	60%
D2H_SMAL	Small	75%

Format Special Bold

You may have noticed that in this manual, every time a new term is introduced and whenever a user is asked to choose a particular option, the term and option are formatted in the same font used for headings. You can apply this same formatting using the **Format Special Bold** command.

You apply this formatting by

1. Highlighting the word or phrase you want to format.
2. Choosing **Special Bold** from the **Format** menu.

Special Bold works as a toggle. Applying it a second time turns special formatting off.

The shortcut key for Special Bold is CTRL+SHIFT+S.

Format Adjust Spacing

You use the **Format Adjust Spacing** to adjust paragraph spacing, one point at a time. While you can access this command by choosing Format and selecting Adjust Spacing, you will probably find it faster to use the line spacing shortcut keys, as shown below:

Pressing These Keys	Performs This Action
CTRL + NUMERIC KEYPAD PLUS	Adds one point of space after the current paragraph
CTRL +ALT + NUMERIC KEYPAD PLUS	Adds one point of space before the current paragraph
CTRL + NUMERIC KEYPAD MINUS	Subtracts one point of space after the current paragraph
CTRL + ALT + NUMERIC KEYPAD MINUS	Subtracts one point of space before the current paragraph

Format Doc-To-Help Markings

You can mark text to be included in and/or excluded from either the manual or the Help file using the Format Doc-To-Help Markings command.

To mark text for the **manual only**, select the text you want to mark and press CTRL+SHIFT+M. The selected text will be formatted in red.

To mark text for **help only**, select the text you want to mark and press CTRL+SHIFT+H. The selected text will be formatted in magenta.

See "View Doc-To-Help Markings" on page 207.

Format Make Into Help

You use the **Format Make Into Help** command to convert your manual into online help. Before Doc-To-Help begins converting the manual to Help, you will be prompted to **Run Document Diagnostics**. See "Tools Doc-To-Help Diagnostics" on page 236 for more information.

D2H_HELP has different style definitions and menu commands from the other Doc-To-Help templates.

If diagnostics have not been run, or diagnostics did not find any errors, Doc-To-Help will save and close the .DOC file, and then save the file with an extension of RTF, associating the RTF file with the template D2H_HELP.DOT.

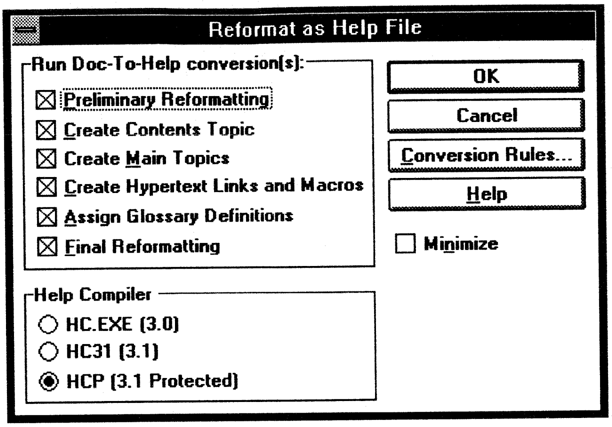
When the saving process is complete, Doc-To-Help will display the **Reformat as Help File** dialog box.

See "Format Reformat as Help" on page 227.

Format Menu (Help–.RTF)

Format Reformat as Help

When you choose **Format Reformat as Help** the Reformat as Help File dialog box will appear, as shown below.



Reformat as Help File

Run Doc-To-Help conversion(s):

☒ Preliminary Reformatting
☒ Create Contents Topic
☒ Create Main Topics
☒ Create Hypertext Links and Macros
☒ Assign Glossary Definitions
☒ Final Reformatting

☐ Minimize

Help Compiler

○ HC.EXE (3.0)
○ HC31 (3.1)
● HCP (3.1 Protected)

The bitmap (.BMP) files and metafiles (.WMF) will be stored in the BITMAPS directory, which can be found below the directory where your manual is saved.

Choosing **Preliminary Reformatting** will reformat symbols and special characters, convert graphics to external bitmap files and metafiles, apply styles from D2H_HELP and strip out text that has been marked for the manual only.

Choosing **Create Contents Topic** will create the contents page that your user first sees when he/she loads your Help file. The contents page is based on your document's table of contents (if you don't have a table of contents, Doc-To-Help will create one for you). You decide how much detail you want to show on the contents page by clicking on the **Conversion Rules** button. (See "Conversion Rules" on page 218.)

If text formatted in Heading 3 style is followed immediately by text formatted in Heading 4 style, Doc-To-Help combines the two into one topic.

Choosing **Create Main Topics** will cause Doc-To-Help to ferret out all text formatted in Heading 2 and 3 styles, and insert a series of footnote codes. These footnote codes make the Headings (and the body text beneath) into topics that can be accessed through online help. Doc-To-Help will make all text formatted in Heading 2, Heading 3 or Heading 4 style a related topic of whatever main topic immediately precedes it.

Choosing **Create Hypertext Links and Macros** will search for Ref and Pageref fields (as well as hypertext link field codes you may have inserted) and convert them into jumps, popups and help macros. Captions will also be converted to popup definitions.

Each Glossary Term will appear in the Help file formatted with a dotted underline.

Choosing **Assign Glossary Definitions** will cause Doc-To-Help to examine your document's Glossary of Terms section (if there is one) and find where in your document these terms appear. When Doc-To-Help finds a match, it will insert the appropriate codes to make your description of that term appear as a popup definition in the Help file. When the user clicks on that term in the Help file the definition will pop up. Doc-To-Help will only create the popup definition for the first term in any given topic.

Choosing **Final Reformatting** will cause Doc-To-Help to remove any unnecessary hidden text, place a hard page break before each topic and save the file in Rich Text Format (RTF).

Turning the **Minimize** option on will reduce Word to an icon as Doc-To-Help converts your manual into help. Reducing Word to an icon will increase the speed of the conversion process by as much as 25% as well as allow you to work on other tasks while Doc-To-Help works in the background. If you restore Word from an icon during the conversion process, you will not be able to switch to other tasks.

Help Compiler

Doc-To-Help installs three different versions of the Microsoft Help Compiler. HC.EXE will create help files that are compatible with the Windows 3.0 help engine (WINHELP.EXE). HC31.EXE and HCP.EXE create help files that are compatible with the Windows 3.1 help engine. HCP.EXE is actually just a special version of HC31.EXE that takes advantage of extended memory. If you will be compiling large help files, or help files with many graphics, you'll probably need to use HCP.EXE.

The Windows 3.1 help engine supports tables, help macros, scalable fonts, non-scrolling regions and sports an improved user interface. Even if you are using Windows 3.0, you can still work with Windows 3.1 help files as long as you use the Windows 3.1 help engine. Doc-To-Help installs the 3.1 version of WINHELP.EXE in your Windows directory if you are using Windows 3.0.

Improving Performance

In addition to improving performance by reducing Word to an icon during the conversion process, you can also add some adrenaline by turning on draft view before initiating the Make Into Help process. You turn draft view on by choosing **Draft** from the **View** menu. WordBasic macros run faster in draft view.

You should also make sure that the entire document "fits" within the left and right screen boundaries meaning that you can see all of your document's left and right margins. If things don't quite fit, choose **Zoom** from the **View** menu and change the magnification to 75%.

How the Windows 3.0 Compiler Handles Tables

The Windows 3.0 Help compiler cannot handle Word for Windows tables. If you are using the 3.0 compiler, Doc-To-Help will convert your tables into attractively formatted, tab-delimited text. The one restriction is that only the *last* column in your table can have word wrapping; the other columns must be wide enough to accommodate the widest lines.

How the Windows 3.1 and HCP Compiler Handle Tables

The Windows 3.1 Compiler recognizes Word for Windows tables. The only drawback is that it does not convert table borders. If you are using the 3.1 compiler, Doc-To-Help will remove the table borders when you make the manual into Help.

Tools Menu (Manuals–.DOC)

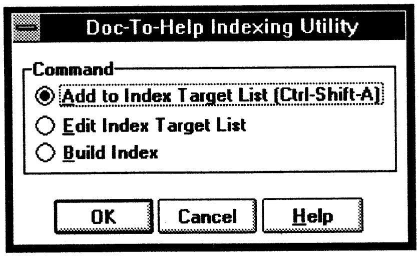
Tools Sort Glossary of Terms

You use the **Tools Sort Glossary of Terms** to alphabetize entries made in the Glossary of Terms section of your document (if your document has such a section). While you don't need to sort these terms for the Assign Glossary Definitions portion of the Help conversion process to work properly, you'll probably want to sort the terms before you print your manual.

Tools Indexing

When you select this command, the **Tools Indexing** dialog box will be displayed as shown below:

For a complete explanation of Doc-To-Help indexing, see "Indexing" on page 65.



Doc-To-Help Indexing Utility

Command

● Add to Index Target List [Ctrl-Shift-A]

○ Edit Index Target List

○ Build Index

Choosing **Add To Index Target List** will let you add selected text to the list of words and phrases Doc-To-Help will search for when you build the index. If **Prompt on Add to Target List** is selected in **Doc-To-Help Options**, the **Index Target Options** dialog box will appear when you choose **Add to Index Target List**. The shortcut key for this command is CTRL+SHIFT+A.

Choosing **Edit Index Target List** will let you set Index Target Options, add, delete or modify tags, or add, delete or modify targets.

Choosing **Build Index** will begin the process of placing index tags at index target locations. Default tags associated with targets defined as

Auto will automatically be placed at the target location. You will be prompted to select which tags are placed at a target defined as Discretionary.

Tools Indexing Add to Index Target List

Note: The limit for index targets is 115 characters.

You use the **Add to Index Target List** command (CTRL+SHIFT+A) to create a list of all words and phrases you want Doc-To-Help to look for when it builds an index. You add a word or phrase by

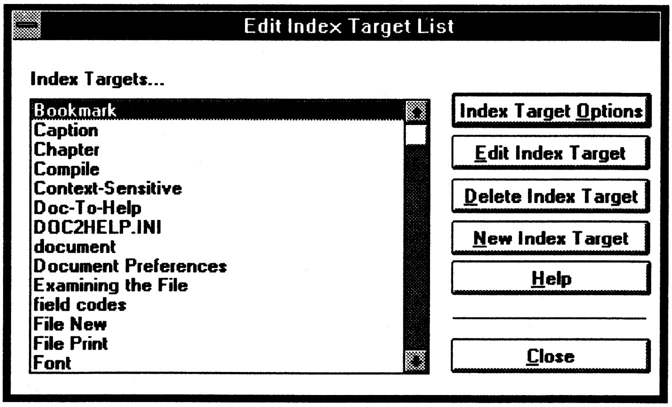
1. Highlighting the word or phrase you want to add.
2. Pressing CTRL+SHIFT+A.

Note: *If Prompt on Add to Index Target List is checked in the Doc-To-Help Options dialog box, Doc-To-Help will display the Index Target Options dialog box whenever you add an index target to the list. (See "Index Target Options Dialog Box" on page 231 for more information.)*

Warning: *Do not use square brackets "[ ]" in index target text.*

Tools Indexing Edit Index Target List

Choosing **Tools Indexing Edit Index Target List** will display the Edit Index Target List dialog box, as shown below.



The list box shows all the index targets in the list. These are the words and phrases Doc-To-Help will look for when building the index.

Pressing **Index Target Options** will display the Index Target Options dialog box. This command will let you decide which index tags will be inserted when the index target is found in the manual, as well as set the target type, search criteria, etc.

Pressing **Edit Index Target** will allow you to change the spelling or phrasing of the selected target.

Pressing **Delete Index Target** will remove the target from the list.

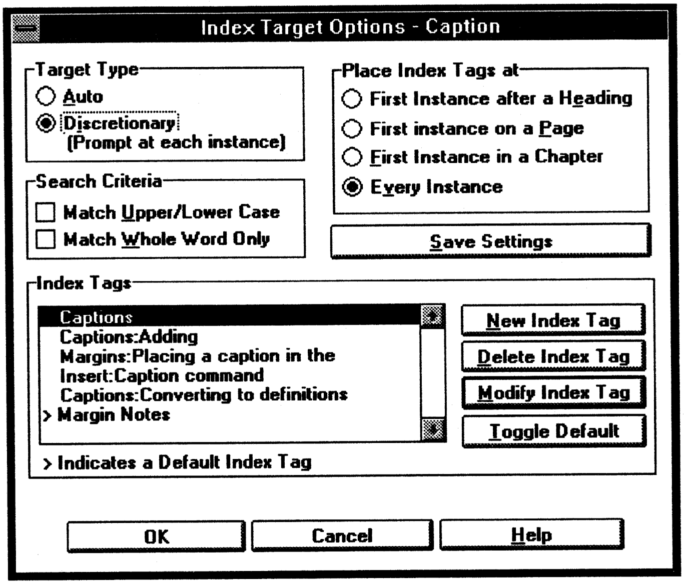
Pressing **New Index Target** allows you to add index targets to the list. This performs the same function as the **Add to Index Target List** command.

Pressing the **Help** button will display Help on editing your index list.

Pressing **Close** will remove the dialog box.

Index Target Options Dialog Box

You determine what Doc-To-Help will do when it finds a particular index target by selecting that item and clicking on the **Index Target Options** button. Clicking on this button will produce a dialog box similar to the one shown below.



Index Target Options - Caption

Target Type
○ Auto
● Discretionary (Prompt at each instance)

Search Criteria
□ Match Upper/Lower Case
□ Match Whole Word Only

Place Index Tags at
○ First Instance after a Heading
○ First instance on a Page
○ First Instance in a Chapter
● Every Instance

Index Tags
Captions
Captions:Adding
Margins:Placing a caption in the
Insert:Caption command
Captions:Converting to definitions
> Margin Notes

> Indicates a Default Index Tag

Please note that the target will be displayed in the title bar of the dialog box.

Auto indexing is the default Target Type.

Clicking **Auto** in the **Target Type** group box will automatically insert default tags at the target location during **Build Index**.

Clicking **Discretionary** in the **Target Type** group box will prompt you to determine which tags will be placed at the index target location during **Build Index**.

In the **Search Criteria** group box you can customize how Doc-To-Help searches for targets. **Match Upper/Lower Case** will find only those instances of the target with the combination of upper and lower case specified by the target list. **Match Whole Word Only** will ignore occurrences that are part of a larger word, e.g. "dog" but not "dogma."

Pressing **Save Settings** will set the defaults (for target type, search criteria, etc.) for any index targets you may add later.

The **Place Index Tags at** option group lets you choose to place index tags at the **First Instance after a Heading, First Instance on a Page, First Instance in a Chapter**, and **Every Instance** of the target.

Pressing **New Index Tag** allows you to associate additional index tags with the selected index target.

Pressing **Delete Index Tag** allows you remove an index tag associated with the selected index target.

Pressing **Modify Index Tag** allows you to change the spelling or phrasing of an index tag.

If the Index Target Type is Auto, make sure the index tags are marked as Default for the Index Target. Otherwise, they'll be ignored.

Pressing **Toggle Default** toggles the default attribute for the selected tag. The way Doc-To-Help handles default tags depends on whether the target type is Auto or Discretionary. If the Index Target Type is Auto, default tags will be automatically placed at the index target location. If the Index Target Type is Discretionary, you will be prompted to place the default tags and/or any non-default tags at the index target location. So, if the Target Type is Discretionary, tags that you want placed at the target location most or all of the time should be designated as default.

Pressing **Help** will display help about the **Index Target Options** dialog box.

Tools Indexing Build Index

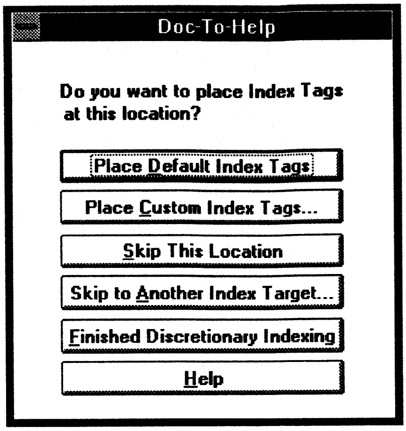
Choosing **Tools Indexing Build Index** will start the process of inserting tags.

Auto Indexing

When you choose **Build Index**, Doc-To-Help will check if there are any targets defined as discretionary in the Index Target List. If all the index targets are defined as Auto, Doc-To-Help will go to each target location and insert Word for Windows index fields representing the default tags for that target. When **Build Index** is complete, the "Indexing is complete" message box will appear.

Discretionary Index Targets

If Discretionary Index Target Type is selected for a particular index target, Doc-To-Help will query you when it finds each occurrence of that index target during the Build Index process, as shown on the next page.


Doc-To-Help
Place Default Index Tags
Place Custom Index Tags...
Skip This Location
Skip to Another Index Target...
Finished Discretionary Indexing
Help

Do you want to place Index Tags at this location?

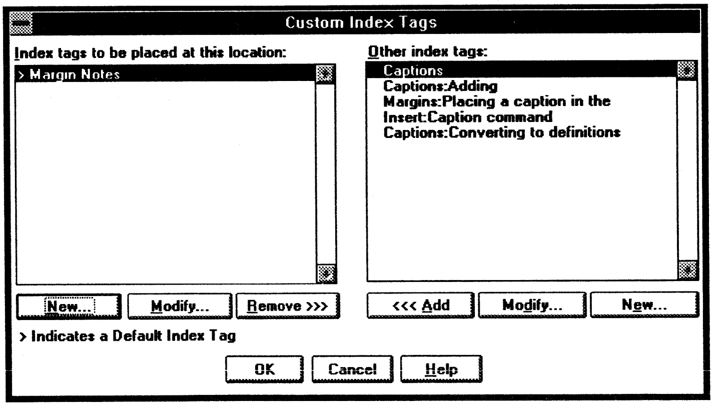
Choosing **Place Default Index Tags** will place index tags marked as default at the current location.

Pressing **Place Custom Index Tags** will display the Custom Index Tags dialog box.

Pressing **Skip this Location** will cause Doc-To-Help to skip to the next occurrence of the current index target.

Pressing **Skip to Another Index Target** allows you to select a different index target to work with. This means that you can interactively build part of the index, take a break, and pick up where you left off without having to review index targets you've already marked.

Custom Index Tags Dialog Box



Index tags to be placed at this location:

> Margin Notes

Other index tags:

Captions
Captions:Adding
Margins:Placing a caption in the
Insert:Caption command
Captions:Converting to definitions

> Indicates a Default Index Tag

Pressing the **Add** button will take one of the "sometimes used" tags in the list box on the right and move it to the list box on the left. The tags in this list will be inserted at the current location.

Pressing the **Remove** button will prevent a particular index tag from being inserted at the current location.

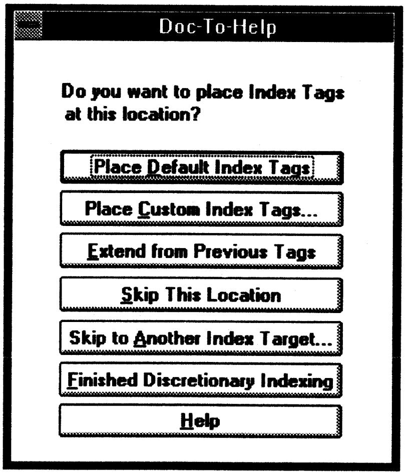
Pressing the **Modify** button will change one of the selected index tags. Modifying an index tag will change the way that entry is inserted throughout the document, not just at the current location. For example, let's say you have the index tag "Marx, Karl" marked as a default entry for the index target "iconoclast." Let's further suppose that half-way through finding all instances of the word "iconoclast," you decide to modify the index tag so that it reads "Marx, Groucho." Any index tags referring to "Marx, Karl" will now refer to "Marx, Groucho." If you want to have index tags referring to *both* Karl and Groucho, use the Add button instead of the Modify button.

Pressing the **New** button allows you to create additional index tags.

Pressing the **OK** button will insert special indexing codes into your document. These codes will be replaced with standard Word for Windows index fields when you finalize your index tags.

What Happens Next

When Doc-To-Help finds the next instance of the index target, you will get an additional button in the dialog box:



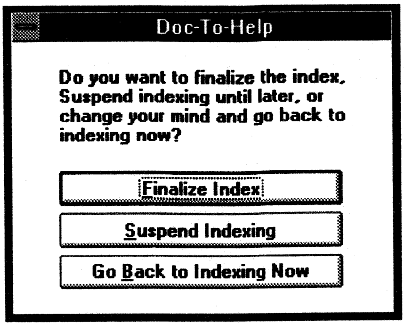
Do you want to place Index Tags at this location?

Doc-To-Help
Place Default Index Tags
Place Custom Index Tags...
Extend from Previous Tags
Skip This Location
Skip to Another Index Target...
Finished Discretionary Indexing
Help

Choosing **Extend From Previous Tags** will extend the index entry from the previous location to the current location. For example, let's say pages 5 through 9 in your manual contain information about "Captions." If these occurrences of the word "caption" are part of the same discussion, you can place the first index tag on page 5. Then when Doc-To-Help finds the occurrence of the word "caption" on page 9 choose **Extend From Previous Tags**. The entry in the index will look like this:

Captions 5-9

Pressing **Finished Discretionary Indexing** indicates that you want to suspend the interactive indexing process. Doc-To-Help will display the Finalize Index dialog box, as shown below.



Doc-To-Help

Do you want to finalize the index, Suspend indexing until later, or change your mind and go back to indexing now?

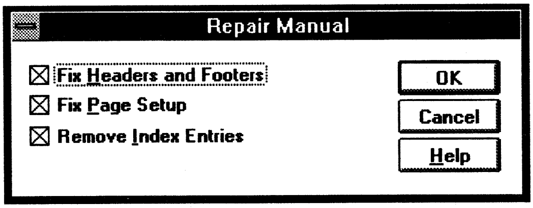
Pressing **Finalize Index** will replace temporary index codes with Word's index field codes. It will also process all index targets marked as Auto. You can see the fruits of your efforts by clicking in the index at the end of your document and pressing the Update Fields key (F9). If this is the first time you are building the index, click in the line that says "Error! No index entries found."

Pressing **Suspend Indexing** leaves everything intact (including the temporary index codes) and gives you the chance to build the index at a later session by choosing **Tools Indexing Build Index**.

Pressing **Go Back to Indexing Now** will take you back to the very first occurrence of the first index target that required prompting and allow you to accept the current tags at that location, make changes or move on to the next index target.

Tools Repair Manual

When you choose this command from a single-file document, the **Tools Repair Manual** dialog box will appear, as shown below.



Repair Manual

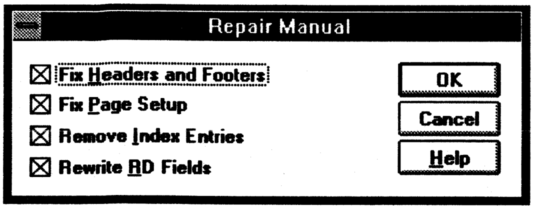
☒ Fix Headers and Footers
☒ Fix Page Setup
☒ Remove Index Entries

Choosing **Fix Headers and Footers** will reset the headers and footers in your manual to match the headers and footers in the attached template.

Choosing **Fix Page Setup** will match the page setup in the manual to the default in the attached template.

Choosing **Remove Index Entries** will remove Word for Windows index fields and Doc-To-Help temporary index codes from the manual.

If you are repairing a multifile project and the master document is active, the **Tools Repair Manual** dialog box will have an additional checkbox for repairing RD fields, as shown below:



☒ Fix Headers and Footers

☒ Fix Page Setup

☒ Remove Index Entries

☒ Rewrite RD Fields

Choosing **Rewrite RD Fields** will delete existing RD fields and update them to reference the correct inside chapter files. RD fields are used to compile the Index and Table of Contents in your master document.

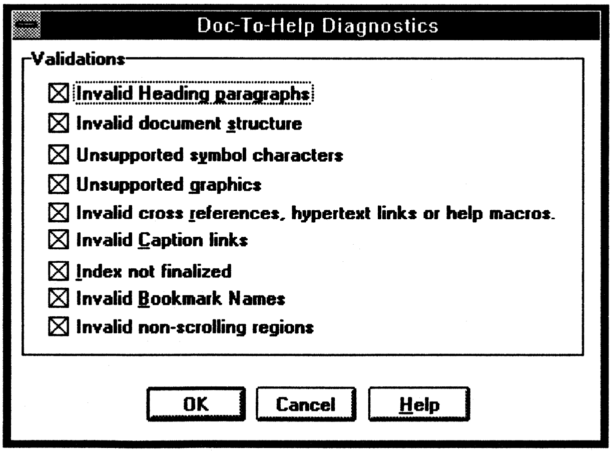
Tools Doc-To-Help Diagnostics

This command will check your manual for any inconsistencies that will cause a problem when converting the document into Help. You can choose this command at any time from the **Tools** menu and you will also be prompted to run it when you choose **Format Make Into Help**.

If errors are found in the manual, Doc-To-Help will create a document called "**D2H_ERR.TXT**" that will be opened when the diagnostic routine finishes. Use this document as a guide to making corrections in the manual. After making corrections you can choose Doc-To-Help Diagnostics again or proceed directly to Make Into Help without checking for errors.

The options checked in the **Doc-To-Help Diagnostics** dialog box are stored in the DOC2HELP.INI file so they'll be the same next time you choose Doc-To-Help diagnostics for that project.

When you choose this command the **Tools Doc-To-Help Diagnostics** dialog box will appear, as shown below:



☒ Invalid Heading paragraphs

☒ Invalid document structure

☒ Unsupported symbol characters

☒ Unsupported graphics

☒ Invalid cross references, hypertext links or help macros.

☒ Invalid Caption links

☒ Index not finalized

☒ Invalid Bookmark Names

☒ Invalid non-scrolling regions

Choosing **Invalid Heading paragraphs** checks if there is at least one Heading 1 paragraph in the manual and, that each Heading 1 paragraph is followed immediately by a paragraph formatted in Heading 2 style. It will also make sure that paragraphs formatted in any of the heading styles are limited to 255 characters.

Choosing **Invalid document structure** will make sure that all document components are present.

Choosing **Unsupported symbol characters** will look for symbol fields that are not formatted in one of the standard Windows fonts. Examples of unsupported fonts include Wingdings and Zapf Dingbats.

Choosing **Unsupported graphics** will flag graphics found in your manual that will not convert to bitmap files or Windows metafiles.

Choosing **Invalid cross references, hypertext links or help macros** will find cross references, link or macro references to bookmarks or headings that no longer exist.

Choosing **Invalid Caption Links** will find links of captions to text that has been deleted.

Choosing **Index not finalized** will check that **Finalize Index** has been run and that all temporary index codes have been converted to index fields.

Choosing **Invalid Bookmark Names** will search for diacritical marks (accents on characters) in bookmark names. The existence of these characters will prevent the Help Compiler from working properly.

Choosing **Invalid non-scrolling regions** will search for paragraphs that should not have the Keep With Next attribute turned on.

A valid non-scrolling region is defined by applying the Keep With Next paragraph formatting option to any number of paragraphs at the beginning of the help topic. However, if there is a paragraph formatted as Keep With Next which appears in the Help topic *after* a paragraph which is not formatted as Keep With Next, the Help Compiler interprets this as an attempt to define a non-scrolling region after a scrolling region, and reports a warning during the compile process.

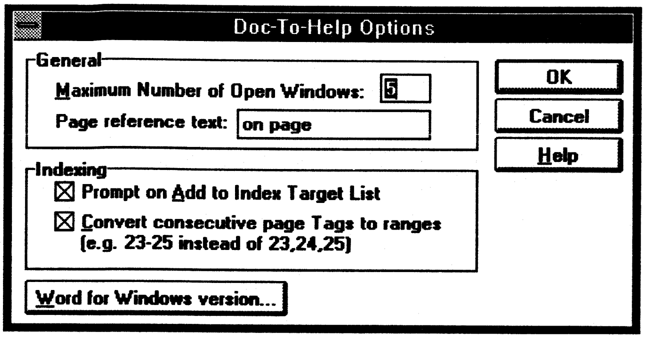
You can use this technique to add Hypertext Links to the non-scrolling region

By default, heading paragraphs are automatically formatted as Keep With Next. This is a property of the heading styles in the D2H_HELP template. You can add one or more paragraphs to the non-scrolling region by selecting the paragraph(s) immediately following the heading paragraph, selecting Paragraph from the Format menu, and clicking the Keep With Next checkbox.

The Invalid Non-Scrolling Region option in Doc-To-Help Diagnostics will report any non-scrolling paragraph (formatted as Keep With Next) which does not follow a heading paragraph, and which comes after a scrolling paragraph (non-Keep With Next).

Tools Doc-To-Help Options

Tools Doc-To-Help Options stores defaults used by Make Into Help and indexing. Choosing this command displays the **Doc-To-Help Options** dialog box, as shown below.



General

Maximum Number of Open Windows: 5

Page reference text: on page

Indexing

☒ Prompt on Add to Index Target List

☒ Convert consecutive page Tags to ranges (e.g. 23-25 instead of 23,24,25)

General

Entering a number for **Maximum Number of Open Windows** lets you set how many documents can be opened during the following Doc-To-Help commands: Build Index, Change Fonts, Doc-To-Help Diagnostics, Repair Manual, Make Into Help, File Summary Info, and File Print. The default value for the maximum number of open windows is 5.

You can change the text that is placed when Doc-To-Help inserts a Pageref field by changing the Page reference text edit box. This feature is useful for people who will be writing manuals in languages besides English, or for people who will be writing in more than one language. *(Note:* This setting is independent of the language version of Word for Windows that you use.)

Indexing

Choosing **Prompt on Add To Target List** will display the **Index Target Options** dialog box when you press **CTRL+SHIFT+A.**

Choosing **Convert consecutive page Tags to ranges** will group references to index tags that appear on consecutive pages. For example, if an index tag appears on pages 5, 6 and 7, the reference numbers in the index will appear as “5-7” rather than “5,6,7.”

Word for Windows Version

This button is available only in the international version of Doc-To-Help and allows you to change which language version of Word for Windows you will be using.

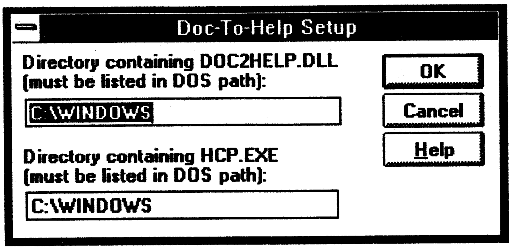
Tools Doc-To-Help Options

See “Tools Doc-To-Help Options” on page 237.

Tools Doc-To-Help

DOC2HELP.DLL performs a number of tasks, including converting pictures contained in your document into disk-based bitmap files and metafiles.

You use the **Tools Doc-To-Help Setup** command to tell Doc-To-Help where to find the Help compiler you’re using and where to find DOC2HELP.DLL.



Directory containing DOC2HELP.DLL (must be listed in DOS path): C:\WINDOWS

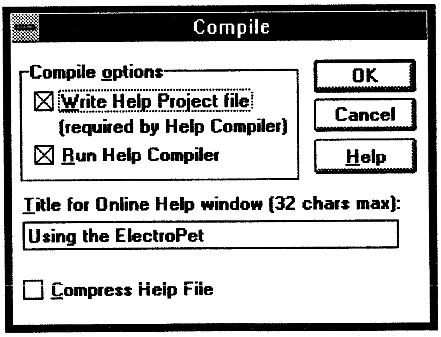
Directory containing HCP.EXE (must be listed in DOS path): C:\WINDOWS

You will need to use this command if you copy or move the Help compiler or DOC2HELP.DLL from the directory where they were installed.

Tools Compile

You use the **Tools Compile** command to compile the RTF file into a Help file. Choosing this command displays the Compile Options dialog box.

Windows 3.0 Compile Options



Compile options

☒ Write Help Project file (required by Help Compiler)

☒ Run Help Compiler

Title for Online Help window (32 chars max): Using the ElectroPet

☐ Compress Help File

When the **Write Help Project file** checkbox is selected, Doc-To-Help will write the Help project file (HPJ file). This text file tells the Help compiler where your documents and bitmaps are saved, what your context string mappings are, the title for the online Help window, and so on. The Help project file has the same name as the RTF file, but with an extension of HPJ.

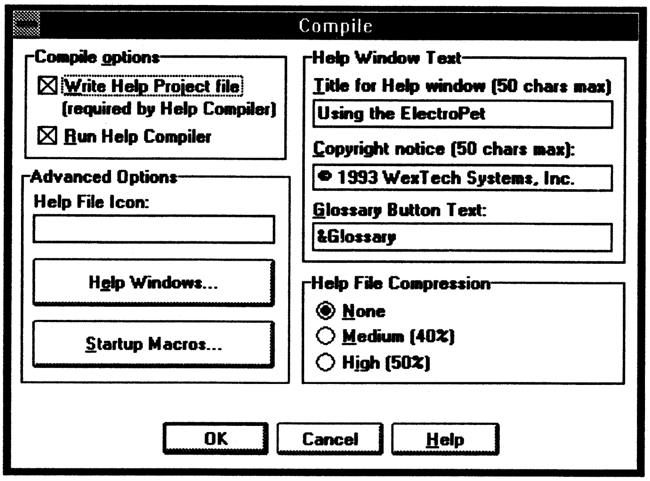
When **Run Help Compiler** is selected, Doc-To-Help will switch to DOS and compile the RTF file into Help. If Doc-To-Help has a problem switching to DOS, you can run the Help compiler yourself by switching to DOS, changing to the directory where the Help file is stored and typing

HC FILENAME.HPJ

By default, the **Title** of the document will appear as the caption for the Help window. You can change this title, if needed.

Choosing **Compress Help File** will make the Help file smaller, but will increase the time it takes to compile.

Windows 3.1 and HCP Compile Options



Compile

Compile options

☒ Write Help Project file (required by Help Compiler)

☒ Run Help Compiler

Advanced Options

Help File Icon:

Help Window Text

Title for Help window (50 chars max)

Using the ElectroPet

Copyright notice (50 chars max):

© 1993 WexTech Systems, Inc.

Glossary Button Text:

&Glossary

Help File Compression

● None

○ Medium (40%)

○ High (50%)

Compile Options

When the **Write Help Project file** checkbox is selected, Doc-To-Help will write the Help project file (HPJ file). This text file tells the Help compiler where your documents and bitmaps are saved, what your context string mappings are, the title for the online Help window, and so on. The Help project file has the same name as the master RTF file, but with an extension of HPJ.

When **Run Help Compiler** is selected, Doc-To-Help will shell to DOS and compile the RTF file into Help. If Doc-To-Help has a problem shelling to DOS, you can run the Help compiler yourself by switching to DOS, changing to the directory where the HPJ file is stored, and typing

HC31 FILENAME.HPJ

for the Windows 3.1 compiler, or

HCP FILENAME.HPJ

for the Windows 3.1 protected-mode compiler.

Advanced Options

There are a variety of icon editing utilities available, including Iconworks which comes with Visual Basic.

You can also specify a custom **Help File Icon**. By default, when you minimize your help file the standard question mark icon will appear. If you're a little tired of this plain yellow icon, you can create your own icon or use shareware icons.

You specify the custom icon you want to use by placing name of the icon in the Help File Icon edit box. If the icon file is not in the same directory as your Doc-To-Help project, you will need to specify the drive and path name as well as the filename.

Pressing the **Help Windows** button accesses the Windows Settings dialog box. For more information, see "Help Windows" on page 223.

Pressing the **Startup Macros** button accesses the Edit Windows Help macros dialog box. For more information, see "Insert Help Macro" on page 211.

Help Window Text

By default, the **Title** of the document will appear as the caption for the Help window. You can change this title, if needed.

Information entered in the Copyright notice edit box will appear when your user chooses **About** from the **WinHelp**'s **Help** menu.

Glossary Button Text

If you are compiling for Windows 3.1 help, Doc-To-Help will add an additional button to the six standard buttons that appear below the Help window menu bar. Clicking this button will jump to your help file's glossary of terms. By default, the text for this button is **Glossary**.

You can create your own buttons using Help macros. For more information, see "Help Macros" on page 149.

Help File Compression

The Windows 3.1 Help compiler offers three compression options. Choosing **None** will compile the RTF file the fastest, but will produce a large Help file. Choosing **Medium** will compile the RTF file less quickly, but will produce a smaller Help file. Choosing **High** will result in the slowest compile time, but the smallest Help file

Tools Run Help

Choosing **Tools Run Help** runs WINHELP.EXE and loads the Help file that was compiled from the active RTF.

Tools Show Context Mapping

A Help file consists of a series of screens called **Help Topics**. The Help compiler assigns a context number to each context string so it can jump to the correct topic. Context-sensitive help is used when you want to display a specific help topic.

To create context-sensitive help you will need to know the relationship between your Help file's context strings and map numbers. You use the **Tools Show Context Mapping** command to create a new document that lists the Topic Title, Context String and Map Number.

This command can only be run after Doc-To-Help has written the Help project file. This file is written when you choose the **Tools Compile** command and has the same name as your document with the extension .HPJ.

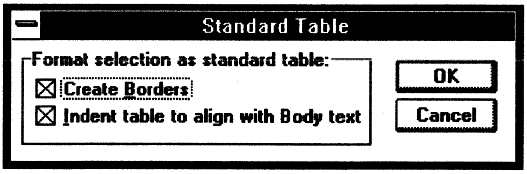
See "Appendix: Accessing Help" on page 245 for examples of adding context-sensitive help to applications.

Table Menu (Manuals–.DOC)

Table Standard Table

You use the Table Standard Table command to insert and/or format tables that line up with body text, are attractively formatted and contain pre-formatted cells. This command will behave differently depending on whether your cursor is already in a table and whether or not any text is selected.

If the cursor is inside a table, the command will help you format the table by displaying the following dialog box.



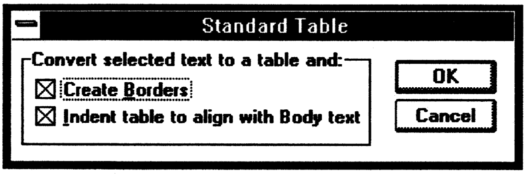
Standard Table

Format selection as standard table:

☒ Create Borders

☒ Indent table to align with Body text

If the cursor is not inside a table and text is selected, the command will help you convert that text into a table and apply formatting by displaying the following dialog box.



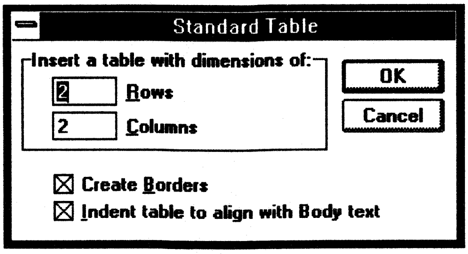
Standard Table

Convert selected text to a table and:

☒ Create Borders

☒ Indent table to align with Body text

If the cursor is not inside a table and no text is selected, the command will present a dialog box asking you how large the table should be, as shown below.



Standard Table

Insert a table with dimensions of:

2 Rows

2 Columns

☒ Create Borders

☒ Indent table to align with Body text

Help Menu (Manuals and Help)

Help Doc-To-Help

Choosing **Help Doc-To-Help** (CTRL+SHIFT+F1) will display help on Doc-To-Help.

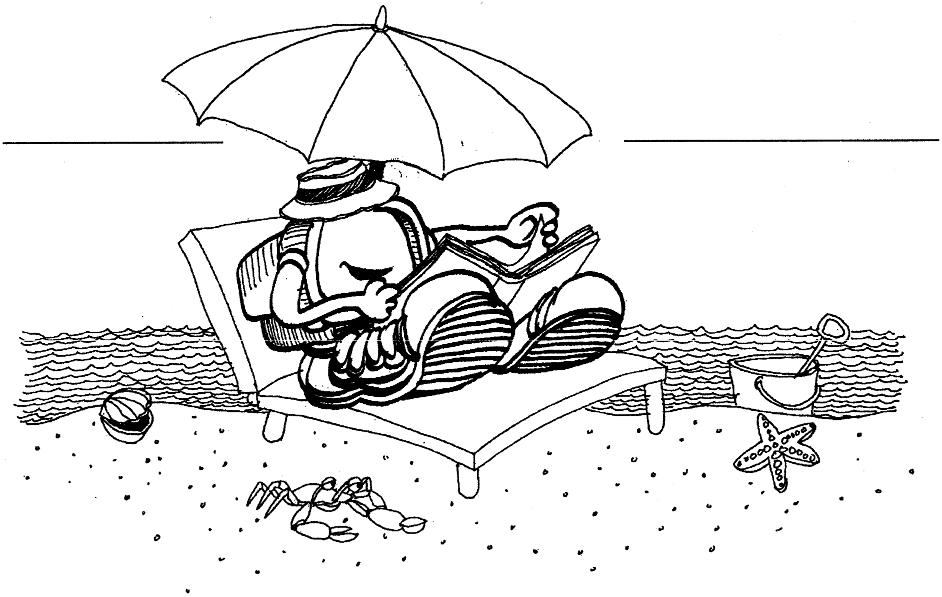
Help Doc-To-Help Keyboard

Choosing **Help Doc-To-Help Shortcut Keys** will display a listing of available shortcut keys. (See "Shortcut Keys" on page 203 for a complete listing.)

Help About Doc-To-Help

Choosing **Help About Doc-To-Help** will display the Doc-To-Help About box which shows which version of Doc-To-Help you're using.

Appendix



Appendix: Accessing Help

Overview

You can access the Help file that you've created with Doc-To-Help from virtually any Windows program that allows calls to the **WinHelp** function. In this section, we'll begin by showing you how to create an icon in Program Manager to call Help. Then you'll get detailed examples for calling Help from WordBasic, Excel, Visual Basic, Access Basic and programs written in C. We'll also show you how to implement context-sensitive help in these applications.

If you're using another application or programming language, you can implement calls to **WinHelp** with some minor modifications. Refer to the documentation for your application or programming language for information on how to access functions in DLLs, specifically functions in the Windows API. Then read the section "Explanation of the WinHelp Function" on page 270 and "Using Help in Applications Written in C" on page 275 for detailed information about the **WinHelp** function.

Calling Help from Program Manager

To Add Help as an Icon in Program Manager

1. Activate the group window to which you want to add an icon.
2. From the **File** menu, choose **New**.
3. In the **New Program Object** dialog box, select the **Program Item** option, and then click **OK**.
4. Type the text to appear below the icon in the **Description** text box.
5. Type the following in the command line text box:

   winhelp.exe [*pathname*]helpfile.hlp

   Substitute the name of your Help file for **helpfile.hlp**. If your file is not in a directory in the DOS path then you will need to enter the pathname as well.

6. If you'd like to add a custom icon, click **Change Icon**. You can add the standard WinHelp icon or choose another icon from a different file.
7. When finished, click **OK**.

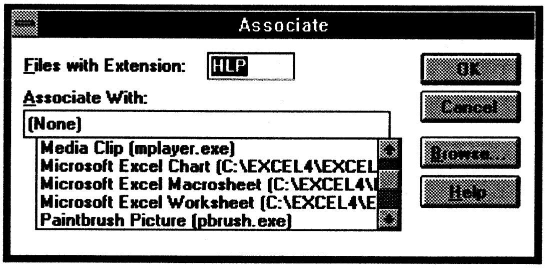
Associating .HLP files with WINHELP.EXE

If from the File Manager you were to double-click on a Word for Windows file, Word for Windows would run and the Word for Windows document you double-clicked would appear on the screen. The same is true for Notepad files, Excel files, etc. If, however, you were to double-click on a Help file (that is, a file with a .HLP extension) you'd probably get a nasty message stating that no application is associated with the file you double-clicked. You'll see this message (instead of the Help file) because, by default, Windows doesn't "know" that .HLP files should be associated with WINHELP.EXE.

You can easily remedy this by using the File Manager's File Associate command.

To Associate .HLP Files with WINHELP.EXE

1. From within the File Manager, select a file with a .HLP extension.
2. From the **File** menu, choose **Associate**. The Associate dialog box will appear, as shown below.



Files with Extension: HLP

Associate With:

(None)

Media Clip (mplayer.exe)
Microsoft Excel Chart (C:\EXCEL4\EXCEL
Microsoft Excel Macrosheet (C:\EXCEL4\
Microsoft Excel Worksheet (C:\EXCEL4\E
Paintbrush Picture (pbrush.exe)

3. In the Associate With edit box, type **WINHELP.EXE**.
4. Click **OK**.

Running Help from Word for Windows

Using WordBasic to Run Help

You can create a simple WordBasic macro to start your online help files. The macro can then be assigned to a menu or a keystroke. If you want your Help file to be available in a specific type of document, then the macro, its menu assignment and keystrokes assignment should be stored in the template the document is based on. For example, if you want to create a Help file for your customized memos, assign the macro and its keyboard and menu assignments to the memo template. If you want your Help file available from any document, the macro and its assignments should be stored in NORMAL.DOT. Refer to your Word for Windows manual for more information on document templates.

To Create a Macro to Run Help

1. Use **File New** to create a document based on the template that will hold the macro and its assignments.
2. From the **Tools** menu choose **Macro**.
3. If you want to access the Help file from all documents, choose **Global** in the **Show** group box. If you want to access the Help file only from the current template, click **Template** in the **Show** group box.
4. Enter a name for the macro in the **Macro Name** text box. Note that macro names can not contain spaces.
5. Click **OK**. The macro editing screen will appear. After the line "Sub MAIN", type

   Shell "winhelp *[pathname]*helpfile.hlp"

   where **helpfile.hlp** is the name of your Help file.

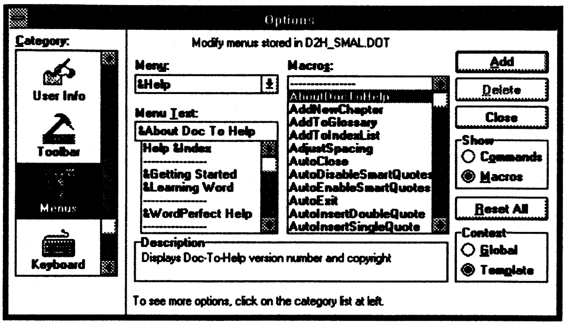
   If your file is not in a directory in the DOS path then you will need to enter the pathname as well.

   The finished code will look like this:

   ```
   Sub MAIN
   Shell "winhelp helpfile.hlp"
   End Sub
   ```
6. From the **File** menu choose **Close**. You will be prompted to save your changes.
7. When prompted for changes, Click **Yes**. The macro edit window will close.
8. From the **File** menu select **Save All**. You will be prompted to save changes to the document template. If you are adding a global macro to the NORMAL.DOT template, you will be prompted to save global glossary and command changes. Click **Yes**.

To Add a Macro to a Menu

1. From the **Tools** menu, choose **Options**.
2. In the **Category** box, select **Menus**. The **Tools Options Menu** dialog will appear as shown below:



Options

Modify menus stored in D2H_SMAL.DOT

Menu: &Help

Menu Text: &About Doc To Help

Macros: AboutDocToHelp

Show: ○ Commands ● Macros

Context: ○ Global ● Template

Description: Displays Doc-To-Help version number and copyright

To see more options, click on the category list at left.

3. In the **Context** box choose **Template** to see a list of the macros that are available only to documents based on the attached template or **Global** to see the macros available to all documents.
4. In the **Show** box choose the **Macros** option button.
5. In the **Menu** box, select the menu to which you want to add the macro.
6. To add an item to a menu, select the macro you want in the **Macros** box. A description of the item appears in the **Description** box. (To add a line separator to the menu, select the dotted line at the top of the list in the **Commands** or **Macros** box.)
7. Type the name in the **Menu Text** box, and then insert an ampersand (&) in front of the letter you want to underline in the name on the menu. For example, type "Memo &Help" to place "Memo Help" on the menu and underline the "H."
8. Choose the **Add** button to add the item to the end of the selected menu.
9. Repeat steps 3 through 8 for each macro or line you want to add.
10. Click **Close**.
11. To save the new menu assignment now, choose **Save All** from the **File** menu. When Word prompts you to save global, glossary, and command changes, or changes to the attached template, click **Yes**.

To Assign a Shortcut Key to a Macro

1. From the **Tools** menu, choose **Options**.
2. In the **Category** box, select **Keyboard**. The **Tools Options Keyboard** dialog box will appear.

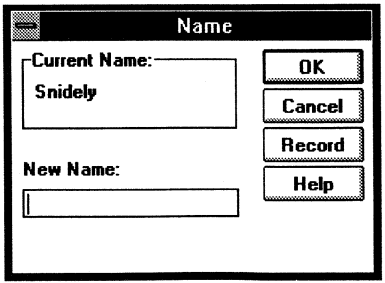
3. In the **Context** box choose **Template** to make the macro available only to documents based on the attached template or Global to make the macro available to all documents.
4. In the **Show** box choose the **Macros** option button.
5. In the **Macros** box, select the macro to which you want to assign a key.
6. Under **Shortcut Key**, select the key combination you want for the macro.

   If an assignment already exists for that key combination, Word displays it to the right of the word **Currently**. If you add a key assignment that Word currently uses for another command, style, or macro, Word assigns the key combination to your choice and removes the previous key assignment. If you do not want to override the previous assignment, type another key combination.
7. Choose the **Add** button to add the item to the end of the selected menu.
8. Repeat steps 3 through 7 for each macro you want to add.
9. Click **Close**.
10. To save the new keystroke assignment now, choose **Save All** from the **File** menu. When Word prompts you to save global, glossary, and command changes, or changes to the attached template, choose **Yes**.

Using WordBasic to Call Context-Sensitive Help

A Help file consists of a series of screens called **Help Topics**. The Help compiler assigns a map number to each topic. You can generate a document containing all the help topics and their corresponding map numbers by using **Tools Show Context Mapping.**

A macro or program can be written to call a specific help topic. The dialog box below is from a sample application. You'll notice that there is a button to access help on the dialog box.



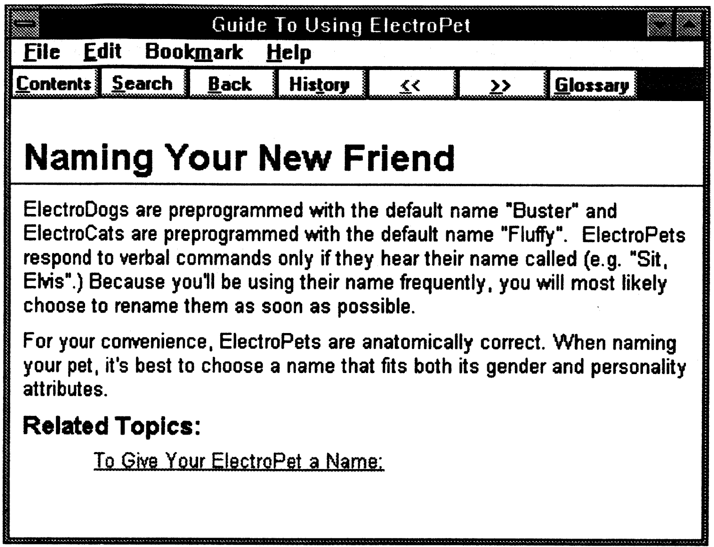
Current Name:
Snidely

New Name:

Sample Dialog Box with Help Button

With context-sensitive help, a help topic specific to the dialog box can be displayed when the user clicks the **Help** button. In this dialog box, the

Help screen for "Naming Your New Friend" will appear when the user clicks on the Help button. That way users don't have to wade through the entire Help file to find the context-sensitive help topic below


Guide To Using ElectroPet
File Edit Bookmark Help
Contents Search Back History << >> Glossary

Naming Your New Friend

ElectroDogs are preprogrammed with the default name "Buster" and ElectroCats are preprogrammed with the default name "Fluffy". ElectroPets respond to verbal commands only if they hear their name called (e.g. "Sit, Elvis".) Because you'll be using their name frequently, you will most likely choose to rename them as soon as possible.

For your convenience, ElectroPets are anatomically correct. When naming your pet, it's best to choose a name that fits both its gender and personality attributes.

Related Topics:

To Give Your ElectroPet a Name:

Sample Help Topic

In this example, the following table was generated with the **Tools Show Context Mapping** command:

Topic Title	Context String	Map Number
Help Contents	HelpContents1	1
Naming Your New Friend	NamingYourNewFriend.2	2
Friends Database	FriendsDatabase.3	3
Companionship	Companionship.4	4
Playfulness	Playfulness.5	5

The second row of the table contains the topic title, context string and map number for the help topic "Naming Your New Friend." Originally formatted in Heading 2 style, the context string appears as a jump from the contents page.

A Sample Macro for Context-Sensitive Help

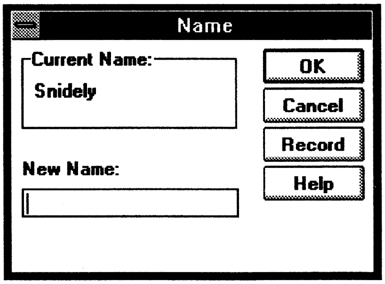
You can create a macro that will display the Help file beginning with the contents topic or can display the Help file beginning with any specific help topic. The subroutine in this macro that displays a specific Help screen can then be called from any other macro.

In this example, we will create two macros. The first is a macro that will display a dialog box to change the name of your pet. Clicking on the Help button will display the Help screen for "Naming Your New Friend." By checking the table generated by **Tools Show Context Mapping**, you can see that the map number for this topic is "2."

Topic Title	Context String	Map Number
Help Contents	HelpContents1	1
Naming Your New Friend	NamingYourNewFriend.2	2
Friends Database	FriendsDatabase.3	3

Clicking the Help button will call a subroutine in a macro called "CallHelp" and pass the map number "2" to the subroutine. This will display the Help screen for "Naming Your New Friend."

This sample dialog box will looks like this:



Current Name: Snidely

New Name:

To Create the Macros for Context-Sensitive Help

1. Open the RTF file created by **Format Make Into Help**.
2. From the **Tools** menu choose **Show Context Mapping.**
3. Determine the map number of the help topic you want to call. In this example, we will be calling help topic "2."
4. Save and/or print it if you want to refer to it later. Close the context map document.
5. Open the template where the macros will be stored.
6. Use **Tools Macro Edit** to create the example dialog box using the following macro code. Call it **Dialog**.

```
Sub MAIN
Begin Dialog UserDialog 320, 144,"Name"
   GroupBox 10, 6, 191, 50,"Current Name:"
   Text 20, 25, 181, 18, "Snidely"   'In 'real life' we'd g
this from data file
   Text 10, 73, 88, 13,"New Name:"
   TextBox 10, 91, 191, 18, .NewName
   OKButton 223, 6, 88, 21
   CancelButton 223, 30, 88, 21
   PushButton 223, 54, 88, 21,"Record"
   PushButton 223, 78, 88, 21,"Help" 'This is the second pu
button
End Dialog

Dim dlg As UserDialog

Again:
Select Case Dialog(dlg)

   Case -1
   REM Code if OK was pressed
   Case 1
   REM Code If First PushButton (Record) was pressed
     .
     .
     .
   Case 2     'If Second Push Button is Pressed
   CallHelp.sContext 2    'Calls CallHelp subroutine sConte
passes
         'Context number 2. This is the number of
         'of the topic screen which will be displayed.
   Goto Again     'Redisplays dialog box
     .
   Case Else
   REM process cancel key
     .
     .
End Select

End Sub
```

7. In this example, pushing the **Help** button will call help topic “2” which is the map number for “Naming Your New Friend” For your own applications, substitute the number “2” with the number of the topic screen you want to call for in the line

   CallHelp.sContext 2

8. Save “Dialog.” Don’t forget to save changes to the template.
9. Use **Tools Macro Edit** to create “CallHelp.” The code is shown below.

```
'-----------------------------------------------------------
'   MACRO:  CallHelp
'   SUBROUTINE:Main
'   PURPOSE:    Calls Sample Help File Contents topic
'-----------------------------------------------------------

Declare Function GetFocus Lib "User"As Integer
Declare Function Winhelp Lib "User" (hWnd As Integer,
lpHelpFile$, wCommand As Integer, dwData As Long) As Integer
Declare Function GetParent Lib "User" (hWnd As Integer) As
Integer
Declare Function GetWindowTask Lib "User" (hWnd As Integer) As
Integer

Sub MAIN
Call sContext 1
End Sub

'-----------------------------------------------------------
'   SUBROUTINE:sContext
'   PURPOSE:    Calls Sample Help File with context number
'-----------------------------------------------------------

Sub sContext(aContextNum)     'aContextNum passed by calling
routine
If AppInfo$(1) >= "Windows 3.1" Then
   'If environment is Windows 3.1, use CallHelp31.sContext
   'Uses 3.1-specific API call for better cursor handling
   Call CallHelp31.sContext aContextNum
   Goto Finish
End If
sCurWindow = GetParent(GetFocus)
sWordHandle = GetWindowTask(GetFocus)
If Winhelp(sCurWindow,"SAMPLE.HLP", 1, aContextNum) Then
   While GetWindowTask(GetFocus) = sWordHandle
       'wait until WinHelp is active before next loop
   Wend
   While GetWindowTask(GetFocus) <> sWordHandle
   'loop until focus returned to Word; prevents Word from
   'returning to the dialog and flashing its title bar
   Wend
End If
Finish:
End Sub
```

10. Use **Tools Macro Edit** to create “CallHelp31.” The code is shown below.

```
'---------------------------------------------------------
'  MACRO:  CallHelp
'  SUBROUTINE:Main
'  PURPOSE:    Calls Sample Help File Contents topic
'---------------------------------------------------------
Declare Function GetFocus Lib "User" As Integer
Declare Function Winhelp Lib "User"(hWnd As Integer, lpHelp
wCommand  As Integer, dwData As Long) As Integer
Declare Function GetParent Lib "User"(hWnd As Integer) As I
Declare Function LoadCursor Lib "User"(hInstance As Integer
lpCursor As  Long) As Integer
'GetCursor is a Windows 3.1-specific API call.
Declare Function GetCursor Lib "USER" As Integer
Declare Sub SetCursor Lib "User"(hCursor As Integer)
Declare Function GetWindowTask Lib "User"(hWnd As Integer)
Integer
Sub MAIN
Call sContext 1
End Sub

'---------------------------------------------------------
'  SUBROUTINE:sContext
'  PURPOSE:    Calls Sample Help File with context number
'---------------------------------------------------------
Sub sContext(aContextNum)
IDC_ARROW = 32512
IDC_WAIT = 32514
sCurWindow = GetParent(GetFocus)
sWordHandle = GetWindowTask(GetFocus)
If Winhelp(sCurWindow, "SAMPLE.HLP", 1, aContextNum) Then
   sWinhelpHandle = GetWindowTask(GetFocus)
   While sWinhelpHandle = sWordHandle
       'help not active yet
       sWinhelpHandle = GetWindowTask(GetFocus)
   Wend
   sArrow = LoadCursor(0, IDC_ARROW)
   sWait = LoadCursor(0, IDC_WAIT)
   If GetCursor = sWait Then SetCursor(sArrow)
   While GetWindowTask(GetFocus) = sWinhelpHandle
       'hang out until focus returns to Word, suppressing
   'the hourglass cursor
       If GetCursor = sWait Then SetCursor(sArrow)
   Wend
End If
End Sub
```

11. In both CallHelp and CallHelp31, substitute your Help file for "SAMPLE.HLP" in the line

    If Winhelp(sCurWindow,"SAMPLE.HLP", 1, aContextNum) Then

12. Close and save "CallHelp" and "CallHelp31." Close and save the template.

13. Create a new document based on the template you just saved. Use **Tools Macro Run** and run "Dialog."

14. Click on the **Help** button in the **Name** dialog. Help will start with the topic screen corresponding to map number "2" (or whatever you substituted).

Make sure the Help file is stored in a directory that's in the DOS path.

Using WinHelp

This macro makes use of the **WinHelp** function which is built into the Windows library of user functions. You can make use of a Windows function with the **Declare** statement. Let's review the declare statement for Help:

```
Declare Function Winhelp Lib "User" (hWnd As Integer,
lpHelpFile$, wCommand As Integer, dwData As Long) As Integer
```

The Declare statement makes the declared function available to WordBasic and describes the parameters required by the function. In **WinHelp** the four required parameters are:

hWnd an integer assigned to the window calling Help;

lpHelpFile$ a string that is the name of your Help file;

wCommand an integer that represents one of six WinHelp commands; and

dwData a long integer that indicates the context topic number.

The words "**as Integer**" at the end of the declare statement indicates that a numeric integer will be returned by this function.

For a more detailed description of the **WinHelp** function, see "Explanation of the WinHelp Function" on page 270.

The Main subroutine of CallHelp simply passes the context number 1 to the subroutine sContext. Context number 1 is always the contents topic (the first topic in the file which was created from the Table of Contents in your manual).

The most efficient use of this macro is to call the subroutine sContext from any other macros where you want context-sensitive help. Simply insert the following line at the point you want to call Help:

MacroName*.sContext *n

where *MacroName* is the macro that contains the sContext subroutine and *n* equals the map number of the topic you want to display. This is especially useful if want to create a Help button in a dialog box.

Much of the complexity of this routine revolves around a peculiarity of Word's custom dialog box. A simpler version of the sContext subroutine would merely call the WinHelp function and exit. If this subroutine is called when the user calls a Help button on a dialog box, the normal handling of the macro controlling the dialog box will be to display the dialog box again after the sContext subroutine ends. There are two problems with this: 1) the sContext routine will continue and end while the user is examining the Help file; and 2) the macro that calls sContext will display the dialog when Word's window is no longer active. When Word is inactive and displays a dialog box, it flashes its window to grab the user's attention. What is in most cases a useful feature is in this case an annoyance. The sContext routine does not return control to the calling macro until the user has either called WinHelp or switched to another application. The sContext routine also changes Word's mouse cursor to the standard hourglass, which Word displays when a macro is running.

Using Excel to Access Help

A Help file consists of a series of screens called **Help Topics**. The Help compiler assigns a map number to each topic. Whenever Help is called from Excel, you must not only specify the Help file name, but the topic number as well. In this sense, Excel help is always "context sensitive."

The general syntax for accessing a help file in Excel is

"Filename.ext!TopicNumber"

where **Filename.ext** is the name of the Help file and **TopicNumber** is either the decimal or hexadecimal number assigned to the help topic you want to access. (Make sure to include the full path of the Help file if it is not in the DOS path.)

Excel will not let you specify a Help file name without a Topic Number. If you wish to call the contents topic of your Help file, you must specify the correct map number for the Contents topic (usually 1). You can generate a document containing all the help topics and their corresponding map numbers using **Tools Show Context Mapping**.

To Display Help in the Course of Running a Macro

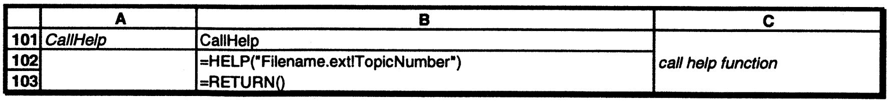
You can create a simple Excel macro to start your online help files. The macro can then be assigned to a menu or a keystroke. Use the help function:

=HELP("Filename.ext!TopicNumber")

Where **Filename.ext** is the name of the Help file (including the full path name to the file if it is not in the DOS path) and **TopicNumber** is the map number of your selected topic. Make sure you include the quotation marks around the name and topic number.

To Create a Macro to Run Help

1. Use **File New** to create a new Macro Sheet. (or open a copy of the structured macro template (Structm.xlt) provided with Excel 4.0).
2. Enter the following code, replace **Filename.ext** with the name of the Help file and **TopicNumber** with the map number for the contents page:



	A	B	C
101	*CallHelp*	CallHelp	
102		=HELP("Filename.ext!TopicNumber")	*call help function*
103		=RETURN()	

3. From the **Edit** menu choose **Define Name...** and name the range B102:B103 "**CallHelp**."

4. From the **File** menu choose **Save**.

Using Excel to Call Context-Sensitive Help

Excel comes with a number of built-in options for calling context-sensitive help from a dialog box or a menu item. Generally, you will include a help reference in the definition of the command, menu or dialog box that requires context-sensitive help. Use **Tools Show Context Mapping** in Doc-To-Help to generate a table of map numbers for your help topics before adding context-sensitive help to your Excel macros.

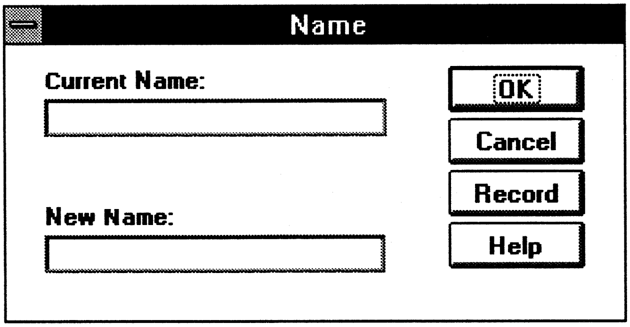
To Generate a Context Map Document

1. Open the RTF file created by **Format Make Into Help**.
2. From the **Tools** menu choose **Show Context Mapping**.
3. Determine the map number of the help topic you want to call.
4. Save and/or print the file if you want to refer to it later. Close the context map document.

To Specify a Help Topic in a Custom Dialog Box

Excel provides a specific **Help** (Item Type 24) button that can be used in a custom dialog box to call context-sensitive help. The help reference to be used when the button is clicked should be entered in the first cell in the item column of the dialog box description for the custom dialog box.

In the following example, the help button in the following dialog box:

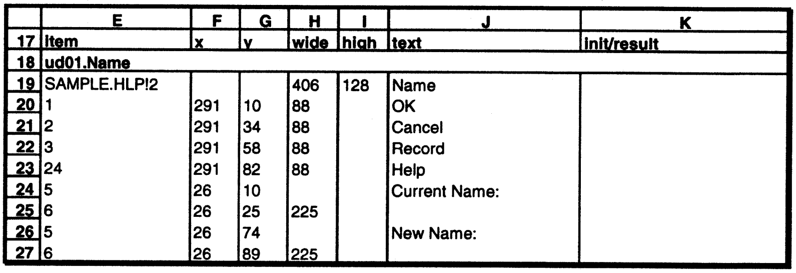


will access the help screen for "Naming Your New Friend." That way, users don't have to wade through the entire help file to find the topic they need.

In this example, the following table was generated with the **Tools Show Context Mapping** command from a Help file called "SAMPLE.HLP."

Topic Title	Context String	Map Number
Help Contents	HelpContents1	1
Naming Your New Friend	NamingYourNewFriend.2	2
Friends Database	FriendsDatabase.3	3

The following dialog box description will produce the sample dialog box shown above. Note that line 23 contains the definition for the help button. Cell E19 contains the help reference that will be accessed when the user clicks on the help button.



	E	F	G	H	I	J	K
17	**Item**	**x**	**y**	**wide**	**high**	**text**	**init/result**
18	**ud01.Name**						
19	SAMPLE.HLP!2			406	128	Name	
20	1	291	10	88		OK	
21	2	291	34	88		Cancel	
22	3	291	58	88		Record	
23	24	291	82	88		Help	
24	5	26	10			Current Name:	
25	6	26	25	225			
26	5	26	74			New Name:	
27	6	26	89	225			

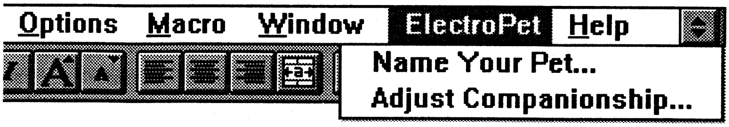
In this case, we are assuming that the Help file will be located in the current directory. If the Help file was located elsewhere, or we weren't sure what the current directory would be when the user clicked on the help button, we would include a full path name. Note that the help reference in a dialog box description does not require quotation marks.

Note: The help reference must be placed in row 1, column 1 of the dialog box definition.

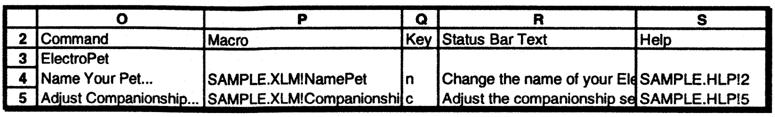
To Specify a Help Topic for a Menu Item

Whenever you use a command table to place a new command on an Excel menu, you have the opportunity to define a custom help reference for that command, simply by entering a help reference in the fifth column of your command table. This technique will work regardless of whether you simply add a single command to a menu or add an entire custom menu to the menu bar. Context-sensitive help for a menu item is accessed either by highlighting the item and pressing **F1** or by pressing **SHIFT+F1** and clicking on the menu item in question.

In the following example, a special menu for ElectroPet commands has been added to the Excel menu bar. Each menu item has its own context-sensitive help topic.



The help references are defined in column 5 of the command table (column S, in this example). Notice that these help references are in the format "Filename.ext!TopicNumber" like all Excel help references.



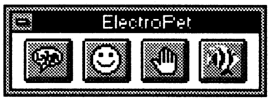
	O	P	Q	R	S
2	Command	Macro	Key	Status Bar Text	Help
3	ElectroPet				
4	Name Your Pet...	SAMPLE.XLM!NamePet	n	Change the name of your Ele	SAMPLE.HLP!2
5	Adjust Companionship...	SAMPLE.XLM!Companionshi	c	Adjust the companionship se	SAMPLE.HLP!5

In this case, we are assuming that the Help file will be located in the current directory. If the Help file was located elsewhere, or we weren't sure what the current directory would be when the user clicked on the help button, we would include a full path name. Note that the help reference in a dialog box description does not require quotation marks.

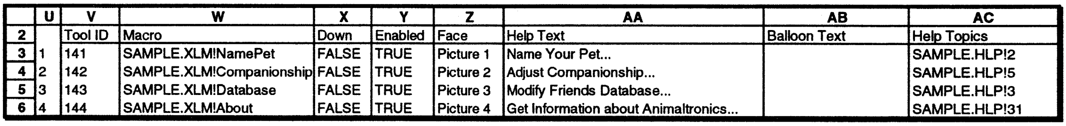
To Add a Help Topic to a Custom Tool

Whenever you use the Excel macro language to place a custom tool on the Toolbar, you have the opportunity to define a custom help reference for that tool, simply by entering a help reference in the eighth column of your tool reference table. Context-sensitive help for a tool is accessed by pressing **SHIFT+F1** and clicking on the tool in question.

In the following example, a special Tool Bar for ElectroPet commands has been added to the Excel menu bar. Each tool has its own context-sensitive help topic.



The help references are defined in column 8 of the tool reference table (column AC, in this example). Notice that these help references are in the format "Filename.ext!TopicNumber" like all Excel help references.



	U	V	W	X	Y	Z	AA	AB	AC
2		Tool ID	Macro	Down	Enabled	Face	Help Text	Balloon Text	Help Topics
3	1	141	SAMPLE.XLM!NamePet	FALSE	TRUE	Picture 1	Name Your Pet...		SAMPLE.HLP!2
4	2	142	SAMPLE.XLM!Companionship	FALSE	TRUE	Picture 2	Adjust Companionship...		SAMPLE.HLP!5
5	3	143	SAMPLE.XLM!Database	FALSE	TRUE	Picture 3	Modify Friends Database...		SAMPLE.HLP!3
6	4	144	SAMPLE.XLM!About	FALSE	TRUE	Picture 4	Get Information about Animaltronics...		SAMPLE.HLP!31

In this case, we are assuming that the Help file will be located in the current directory. If the Help file was located elsewhere, or we weren't sure what the current directory would be when the user clicked on the help button, we would include a full path name. Note that the help reference in a tool reference table does not require quotation marks.

Using Microsoft Access to Call Help

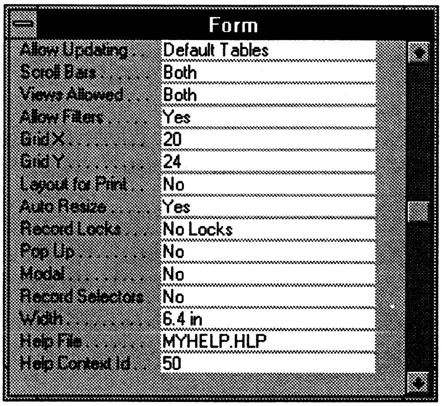
Microsoft Access is designed for extremely easy integration of help into application screens. You can program context-sensitive help for your application by setting a **Help File** property for a form and **Help Context ID** property for the form and its controls.

However, if you want to add a help button or help menu item to a form, the process is slightly more involved. Instructions are included here for programming a context-sensitive help call from a help button, using AccessBasic. For a more thorough discussion of the **WinHelp**function and various ways to call help, see "Explanation of the WinHelp Function" on page 270.

To Program Context-Sensitive Help in Microsoft Access

1. Open the form in Design View
2. Display the form's Property Sheet.

Click the Properties button in the toolbar to display the Property Sheet



Form

Allow Updating . . .	Default Tables
Scroll Bars	Both
Views Allowed . . .	Both
Allow Filters	Yes
Grid X	20
Grid Y	24
Layout for Print . . .	No
Auto Resize	Yes
Record Locks	No Locks
Pop Up	No
Modal	No
Record Selectors .	No
Width	6.4 in
Help File	MYHELP.HLP
Help Context Id . . .	50

You should make sure that the help file is located either in the Windows directory or another directory that is included in the path.

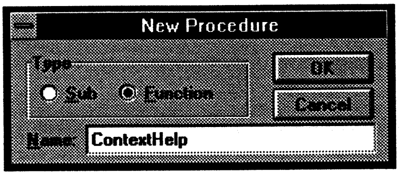
3. Scroll down to the **Help File** property and set its value to the name of your help file.
4. Set the **Help Context ID** property to the screen that should be displayed when the user presses **F1**.
5. If certain controls on the form should display help screens other than the one specified for the form, set the **Help Context ID** property of each control to the appropriate context number.

Including Help Buttons on Microsoft Access Forms

1. Create a new Module, or open an existing Module, and insert the following in the **Declarations** section:

```
Declare Function WinHelp Lib "User" (ByVal hWnd As Integer, ByVal
lpHelpFile As String, ByVal wCommand As Integer, dwData As Any)
As Integer
Global Const HELP_CONTEXT = 1        'Dsply topic by context number
Global Const HELP_QUIT = 2           'Terminate help
Global Const HELP_INDEX = 3          'Display Contents Screen
Global Const HELP_CONTENTS = 3       'Display Contents Screen
Global Const HELP_HELPONHELP = 4     'Display help on using help
Global Const HELP_SETINDEX = 5       'Redefine Contents Screen
Global Const HELP_SETCONTENTS = 5    'Redefine Contents Screen
Global Const HELP_CONTEXTPOPUP = 8   'Display topic in Popup Window
Global Const HELP_FORCEFILE = 9      'Ensure Help file is loaded
Global Const HELP_KEY = 257          'Search alt. keyword table
Global Const HELP_COMMAND = 262      'Execute a Help macro
Global Const HELP_PARTIALKEY = 261   'Call search engine in winhelp
Global Const HELP_SETWINPOS = 515    'Set Help window position
```

2. Using the **New Procedure...** command from the **Edit** menu, create a new function named **ContextHelp**.



Type: ○ Sub ● Function

Name: ContextHelp

3. Insert the following code:

```
Function ContextHelp (aHelpFile As String, aContext As Long)
x = WinHelp(hWnd, aHelpFile, HELP_CONTEXT, ByVal aContext)
End Function
```

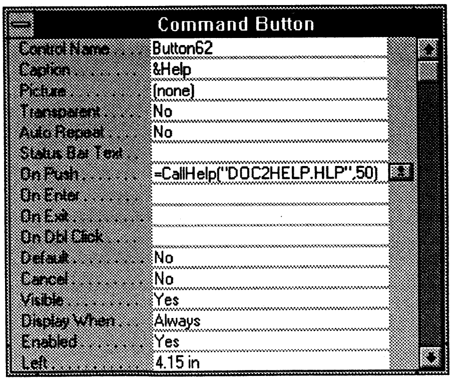
4. Open the form in Design View.

5. Add a Push Button to the form. This will be the help button, so you should set its Caption property to “&Help,” or something to that effect.

6. Set its On Push property to the following value:

   =ContextHelp(*“Helpfile”, ContextNum*)

   where *Helpfile* is the name of the help file to display upon pushing the button, and *ContextNum* is the context number of the desired help screen.



Control Name	Button62
Caption	&Help
Picture	(none)
Transparent	No
Auto Repeat	No
Status Bar Text	
On Push	=CallHelp("DOC2HELP.HLP",50)
On Enter	
On Exit	
On Dbl Click	
Default	No
Cancel	No
Visible	Yes
Display When	Always
Enabled	Yes
Left	4.15 in

Using Visual Basic to Call Help

Visual Basic provides several different ways to program help into your application. The simplest of these is a "plug and play" interface which lets you specify a help file and context numbers just by setting properties in your application. This provides context-sensitive help when the user presses the **F1** key.

If you need to include help buttons and menu commands in your application, the Common Dialog control available in Visual Basic 3.0 (and the professional version of VB 2.0) gives you easy access to the six most commonly uses help commands.

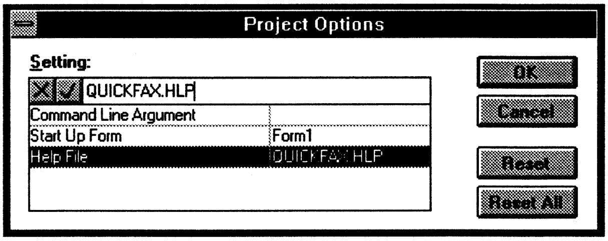
Finally, if you want to access the complete command set of the WinHelp function, you can use it directly in your application code by declaring WinHelp, along with other functions, structures and constants, in a code module.

Programming Context-Sensitive Help in Visual Basic

Visual Basic versions 2.0 and later include a highly automated method of calling context-sensitive help. This involves setting a **HelpFile** property of the **App** object, and specifying a **HelpContextID** property for forms and controls.

To Program Context-Sensitive Help in Visual Basic

1. From the **Options** menu, choose **Project...**
2. Set the **Help File** property to the name of the help file for your application



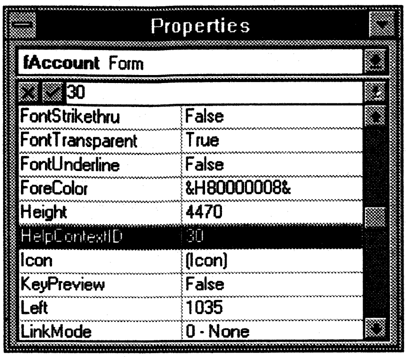
Project Options

Setting:

QUICKFAX.HLP

Command Line Argument	
Start Up Form	Form1
Help File	QUICKFAX.HLP

3. For each form in your application, set the **ContextID** property for the form to the context number of the screen you want to display when the user presses the **F1** key.



Properties

fAccount Form

30

FontStrikethru	False
FontTransparent	True
FontUnderline	False
ForeColor	&H80000008&
Height	4470
HelpContextID	30
Icon	(Icon)
KeyPreview	False
Left	1035
LinkMode	0 - None

4. If you want to display different help screens depending on which control has the focus, set the **HelpContextID** property for each control to the appropriate value.

Calling Help with the Common Dialog Control

Visual Basic's Common Dialog custom control, in addition to displaying common dialog boxes, can manipulate help for your application. The properties used for displaying help are the **HelpCommand**, **HelpContext**, **HelpFile**, **HelpKey** and **Action** properties. By setting these properties of the Common Dialog control, your application can open and close help for your application, display context-sensitive help, redefine the contents screen, and even invoke a keyword search.

Before you use the Common Dialog control, you should define the WinHelp command values as global constants in a code module.

To Define the WinHelp Commands

1. Create a new module or open an existing module in your application.
2. Insert the following into the Declarations section of the module:

```
Global Const HELP_CONTEXT = 1 'Display topic by context number
Global Const HELP_QUIT = 2    'Terminate help
Global Const HELP_INDEX = 3   'Display Contents Screen
Global Const HELP_CONTENTS = 3    'Display Contents Screen
Global Const HELP_HELPONHELP = 4 'Display help on using help
Global Const HELP_SETINDEX = 5    'Redefine Contents Screen
Global Const HELP_SETCONTENTS = 5 'Redefine Contents Screen
Global Const HELP_CONTEXTPOPUP = 8    'Disp. topic in Popup Window
Global Const HELP_FORCEFILE = 9   'Ensure Help file is loaded
Global Const HELP_KEY = 257   'Search alternate keyword table
Global Const HELP_COMMAND = 262   'Execute a Help macro
Global Const HELP_PARTIALKEY = 261    'Call winhelp search engine
Global Const HELP_SETWINPOS = 515 'Set Help window position
```

To Display the Contents Screen

1. Define the WinHelp commands in a code module, if you haven't already done so.
2. Add a Common Dialog control to a form in your application, if it doesn't already have one.

3. In the Code Window, open the event procedure that you want to display help.
4. Type a command to set the Common Dialog control's **HelpFile** property to the name of the help file you want to display.
5. Type a command to set the Common Dialog control's **HelpCommand** property to HELP_CONTENTS.
6. Type a command to set the Common Dialog control's **Action** property to 6.

The following example displays the Contents Screen in MYHELP.HLP using the Common Dialog control CMDialog1 on the form AppForm.

```
Sub Command1_Click ()
    AppForm.CMDialog1.HelpFile = "MYHELP.HLP"
    AppForm.CMDialog1.HelpCommand = HELP_CONTENTS
    AppForm.CMDialog1.Action = 6
End Sub
```

To Display Help on Help

1. Define the WinHelp commands in a code module, if you haven't already done so.
2. Add a Common Dialog control to a form in your application, if it doesn't already have one.
3. In the Code Window, open the event procedure that you want to display help.
4. Type a command to set the Common Dialog control's **HelpCommand** property to HELP_HELPONHELP.
5. Type a command to set the Common Dialog control's **Action** property to 6.

The following example displays the help on help using the Common Dialog control CMDialog1 on the form AppForm.

```
Sub Command1_Click ()
    AppForm.CMDialog1.HelpCommand = HELP_HELPONHELP
    AppForm.CMDialog1.Action = 6
End Sub
```

To Display a Help Topic by Context Number

1. Define the WinHelp commands in a code module, if you haven't already done so.
2. Add a Common Dialog control to a form in your application, if it doesn't already have one.
3. In the Code Window, open the event procedure that you want to display help.
4. Type a command to set the Common Dialog control's **HelpFile** property to the name of the help file you want to display.
5. Type a command to set the Common Dialog control's **HelpCommand** property to HELP_CONTEXT.

6. Type a command to set the Common Dialog control's **HelpContext** property to the context number identifying the help topic you want to display.
7. Type a command to set the Common Dialog control's **Action** property to 6.

The following example displays the topic mapped to context number 4 in MYHELP.HLP using the Common Dialog control CMDialog1 on the form AppForm.

```
Sub Command1_Click ()
    AppForm.CMDialog1.HelpFile = "MYHELP.HLP"
    AppForm.CMDialog1.HelpCommand = HELP_CONTEXT
    AppForm.CMDialog1.HelpContext = 4
    AppForm.CMDialog1.Action = 6
End Sub
```

To Close Help

1. Define the WinHelp commands in a code module, if you haven't already done so.
2. Add a Common Dialog control to a form in your application, if it doesn't already have one.
3. In the Code Window, open the event procedure that you want to close help.
4. Type a command to set the Common Dialog control's **HelpFile** property to the name of the help file previously displayed.
5. Type a command to set the Common Dialog control's **HelpCommand** property to HELP_QUIT.
6. Type a command to set the Common Dialog control's **Action** property to 6.

The following example closes the help file MYHELP.HLP using the Common Dialog control CMDialog1 on the form AppForm.

```
Sub Command1_Click ()
    AppForm.CMDialog1.HelpFile = "MYHELP.HLP"
    AppForm.CMDialog1.HelpCommand = HELP_QUIT
    AppForm.CMDialog1.Action = 6
End Sub
```

To Redefine the Contents Screen

1. Define the WinHelp commands in a code module, if you haven't already done so.
2. Add a Common Dialog control to a form in your application, if it doesn't already have one.
3. In the Code Window, open the event procedure that you want to redefine the contents screen.
4. Type a command to set the Common Dialog control's **HelpFile** property to the name of the help file whose Contents Screen you want to redefine.

5. Type a command to set the Common Dialog control's **HelpCommand** property to HELP_SETCONTENTS.
6. Type a command to set the Common Dialog control's **HelpContext** property to the context number identifying the help topic you want to designate as the Contents Screen.
7. Type a command to set the Common Dialog control's **Action** property to 6.

The following example designates the topic mapped to context number 4 in MYHELP.HLP as the Contents Screen, using the Common Dialog control CMDialog1 on the form AppForm.

```
Sub Command1_Click ()
    AppForm.CMDialog1.HelpFile = "MYHELP.HLP"
    AppForm.CMDialog1.HelpCommand = HELP_SETCONTENTS
    AppForm.CMDialog1.HelpContext = 4
    AppForm.CMDialog1.Action = 6
End Sub
```

To Search Help for a Keyword

1. Define the WinHelp commands in a code module, if you haven't already done so.
2. Add a Common Dialog control to a form in your application, if it doesn't already have one.
3. In the Code Window, open the event procedure that you want to search help.
4. Type a command to set the Common Dialog control's **HelpFile** property to the name of the help file you want to search.
5. Type a command to set the Common Dialog control's **HelpCommand** property to HELP_KEY.
6. Type a command to set the Common Dialog control's **HelpKey** property to the key word or phrase you want to search for in Help.
7. Type a command to set the Common Dialog control's **Action** property to 6.

The following example searches for the keyword “Introduction” in MYHELP.HLP using the Common Dialog control CMDialog1 on the form AppForm.

```
Sub Command1_Click ()
    AppForm.CMDialog1.HelpFile = "MYHELP.HLP"
    AppForm.CMDialog1.HelpCommand = HELP_KEY
    AppForm.CMDialog1.HelpKey = "Introduction"
    AppForm.CMDialog1.Action = 6
End Sub
```

Using the WinHelp Function in Visual Basic

You can take full advantage of the WinHelp function by calling it directly from your Visual Basic programs. Your application can search the Help file for a key word, redefine the Contents screen, and even perform searches using an alternate keyword list.

However, this approach is less direct and more error prone than the other two methods. (See "Programming Context Sensitive Help in Visual Basic" on page 262 and "Calling Help with the Common Dialog Control" on page 263.) If you're using an old version of Visual Basic that doesn't support these methods, we recommend upgrading to the latest version.

We've included the declarations you need to make and some sample code. You'll want to read the section "Explanation of the WinHelp Function" on page 270 if you haven't already done so, and we strongly suggest that you refer to the sections in the Visual Basic documentation on calling procedures in DLLs (chapter 24 in the *Visual Basic 3.0 Programmer's Guide*).

Declarations in the Code Module

To use the WinHelp function, you'll need to declare it in a code module. Start by creating a new module, or opening an existing one, and make sure that the Declarations section is displayed.

1. Place the following Declare statement in the module. *Make sure* that the declaration appears on a single line.

```
Declare Function WinHelp Lib "User" (ByVal hWnd As Integer, ByVal
lpHelpFile As String, ByVal wCommand As Integer, dwData As Any)
As Integer
```

2. Declare the following global constants in the module:

```
Global Const HELP_CONTEXT = 1 'Disp. topic by context number
Global Const HELP_QUIT = 2    'Terminate help
Global Const HELP_INDEX = 3   'Display Contents Screen
Global Const HELP_CONTENTS = 3    'Display Contents Screen
Global Const HELP_HELPONHELP = 4 'Display help on using help
Global Const HELP_SETINDEX = 5    'Redefine Contents Screen
Global Const HELP_SETCONTENTS = 5 'Redefine Contents Screen
Global Const HELP_CONTEXTPOPUP = 8   'Disp. topic in Popup Window
Global Const HELP_FORCEFILE = 9   'Ensure Help file is loaded
Global Const HELP_KEY = 257    'Search alternate keyword table
Global Const HELP_COMMAND = 262   'Execute a Help macro
Global Const HELP_PARTIALKEY = 261    'Call winhelp search engine
Global Const HELP_SETWINPOS = 515 'Set Help window position
```

3. Declare the following Type definition in the module:

```
Type MULTIKEYHELP
    mkSize As Integer
    mkKeylist As String * 1
    szKeyphrase As String * 253
End Type
```

Examining the WinHelp Declararation

Take a moment to examine the **WinHelp** function as it is declared in the module:

```
Declare Function WinHelp Lib "User" (ByVal hWnd As Integer,
ByVal lpszFilename As String, ByVal wCmd As Integer, dwData
As Any) As Integer
```

The first three parameters are declared **ByVal**, but for different reasons. By default, Visual Basic passes arguments by reference. The **WinHelp** function expects a value argument for *hWnd* and *wCmd*. Specifying **ByVal** in the declaration ensures that values, rather than addresses, will be passed.

The *lpszFileName* argument represents a text string designating a filename and path, if necessary, for a Help file. It is specified as **ByVal**, which ensures that the value passed will be the address of the string data, rather than the address of the string descriptor.

The *wCmd* parameter specifies an action to be performed with the help file. Again it is passed by value, so that the actual help command constant, and not the address of an integer variable, is passed.

The *dwData* parameter is used to specify additional information that WinHelp needs to execute the command specified in the *wCmd* parameter. Since the format and data type of *dwData* depend upon the value of *wCmd* passed when your application calls **WinHelp,** the *dwData* parameter is specified ***As Any*** in the Basic declaration. This allows for any data type to be passed. In most calls to **WinHelp** you pass the *dwData* argument with the **ByVal** specification right in the call, as shown in the following fragment:

TempNum% = WinHelp(hWnd, HelpFile, fuCommand, ByVal dwData)

Depending on the help command specified in the fuCommand parameter, *dwData* in the preceding line could be a pointer to a text string, indicating a keyword to look up, or a **Long** integer indicating the context number identifying a specific topic.

When the last parameter must be passed by value, the ByVal keyword is used in the command line. When the last parameter is to be passed by reference, ByVal is left out.

The HELP_MULTIKEY command is used to specify a topic listed in an alternate keyword table. The function is used as follows, where **Multikey** is the data structure for passing the keyword table and keyword (the **Multikey** data structure should be declared in a module).

TempNum% = WinHelp(hWnd, HelpFile, fuCommand, Multikey)

Calling the Help File (HELP_INDEX)

You can call the WinHelp function within an event procedure for a menu item or control. Make sure that you have the following declarations on the form level:

```
' Form-level variables
Dim HelpCmd As Integer
Dim CallHelpFile As String

' Constants for Context Sensitive Help
Const CallHelpFile = "MYFILE.HLP" 'Sets help file
'
```

Substitute the name of your Help file for "MYFILE.HLP." If the file is not in the DOS or network path you will need to define the path explicitly. Then, simply make sure that the following statement is in the event procedure in the control or menu item that will activate Help:

Temp% = WinHelp(hWnd, CallHelpFile, HELP_INDEX, ByVal curData$)

Note: CurData$ is ignored by this command.

Calling Context-Sensitive Help (HELP_CONTEXT)

For additional methods of calling context-sensitive help see "Using Help in Applications Written in C" on page 275.

The following is a simple subroutine that can be used to call context-sensitive help. Simply pass the desired context topic number from any control or menu item which indicates help for a specific topic. The most efficient way to work with topic numbers is to declare constants on the form or global level.

```
Sub ContextHelp (Context As Long)
    HelpCmd = HELP_CONTEXT    ' Set command for contexts

    ' Call WinHelp with context number
    Temp% = WinHelp(hWnd, CallHelpFile, HelpCmd, ByVal
Context)

End Sub
```

The calling routine would look something like this:

```
Sub CatPic_Click()
   ContextHelp(rgnElectroCat)
End Sub
```

In this example, users will get the help topic about the ElectroCat when they click on the CatPic picture control. Note that the constant **rgnElectroCat** instead of an explicit topic number is passed to the **ContextHelp** function. The topic number for "ElectroCat" was assigned to the constant in the form declarations. Assigning topic numbers to constants at the form level will make your Visual Basic application easier to maintain. If the topic numbers change you'll only need to update the declarations.

Canceling Help

Windows Help requires an application to explicitly cancel Help so that Windows Help can free any resources it used to keep track of the application and its Help files. The application can do this at any time; well-behaved Windows applications cancel Help as they exit.

You must call the **WinHelp** function and specify the HELP_QUIT value, as shown in the following example:

TempNum% = WinHelp(hWnd, CallHelpFile, HELP_QUIT, 0)

If the application has made any calls to the **WinHelp** function, it must cancel Help before it closes its main window (for example, in response to the WM_DESTROY message in the main window procedure). An application needs to call **WinHelp** only once to cancel Help, no matter how many Help files it has opened. Windows Help remains running until all applications or dynamic link libraries that have called the **WinHelp** function have canceled Help.

Explanation of the WinHelp Function

Windows' integrated support for on-line help is implemented with a versatile function called WinHelp. The WinHelp function, which lives in the USER.EXE library, is the application programming counterpart to the WINHELP.EXE help engine. As a part of the Windows API (application programming interface), it is responsible for starting and quitting the help engine, loading help files, and displaying the appropriate help screen. As we'll see, the WinHelp function has some other, more esoteric tricks up its sleeve as well.

Some knowledge of the WinHelp function is necessary if you need to program context-sensitive help into an application that doesn't let you plug help filenames and context numbers into its objects, or if you need to go beyond the "plug-and-play" context-sensitive help provided by tools like Microsoft Access and Visual Basic.

WinHelp Syntax

Here is the syntax of the WinHelp function, as it is declared in C.

```
BOOL WinHelp(hWnd, lpszFileName, fuCommand, dwData)
HWND hWnd; /* handle of window requesting help
LPCSTR lpszFileName;  /* address of directory-path string
UINT fuCommand;    /* type of help
DWORD dwData;  /* additional data
```

Let's take a moment to analyze this code to see what it tells us about the WinHelp function. Starting with the data type BOOL, this tells us that the function will return an integer value of one or zero. In fact, any non-zero return value indicates that WinHelp was successful in whatever we asked it to do. Zero indicates that an error occurred.

Next is the calling syntax, which specifies the name of the function and the receiving variable names which assume the values of the arguments passed to the function.

```
    WinHelp(hWnd, lpszFileName, fuCommand, dwData)
```

This tells us that there are four arguments to the Winhelp function. The next four lines tell us the data types of the arguments, and give some clues about the role of each argument.

```
    HWND hWnd;              /* handle of window requesting help    */
    LPCSTR lpszFileName;    /* address of directory-path string    */
    UINT fuCommand;         /* type of help                        */
    DWORD dwData;           /* additional data                     */
```

We'll discuss the meaning of these arguments in detail.

The Windows Handle Parameter (hWnd)

The *hWnd* parameter is an integer which identifies the window requesting Help. The Windows Help application uses this identifier to keep track of which applications have requested Help. Every application window has a handle that can be passed in this position.

The Help Filename Parameter (lpszFileName)

The *lpszFileName* argument is a long pointer to a text string designating a filename and path, if necessary, for a Help file.

The Windows Command Parameter (fuCommand)

The *fuCommand* parameter is an unsigned integer specifying an action to be performed with the help file. It may be set to any of the values listed below:

Global Constant	Value	Description
HELP_CONTEXT	1	Displays a specific help topic identified by a Long Integer number specified in the dwData parameter.
HELP_QUIT	2	Notifies the Help application that the specified Help file is no longer in use.
HELP_INDEX	3	Displays the Contents screen of the specified Help file, as designated by the author of the Help file. The dwData parameter is ignored.
HELP_HELPONHELP	4	Displays help for using the WinHelp application itself. The dwData parameter is ignored.
HELP_SETINDEX	5	Redefines the Contents screen as the help topic whose context number is specified by the dwData parameter.
HELP_KEY	257	Displays the first corresponding topic found by a search for the keyword specified by the dwData parameter.
HELP_MULTIKEY	513	Displays help for a keyword found in an alternate keyword table. The dwData parameter is a far pointer to a special data_structure containing the size of the string, the letter of the alternate table, the keyword string, and a null character to terminate the string.

In addition to the values above, Windows 3.1 supports the following values:

Global Constant	Value	Description
HELP_CONTENTS	1	Same as HELP_INDEX.
HELP_SETCONTENTS	5	Same as HELP_SETINDEX.
HELP_CONTEXTPOPUP	8	Displays a topic, corresponding to a specified context number, in a popup window.
HELP_PARTIALKEY	261	Searches the keyword list in the help file for a specified keyword string. If one topic exists that includes the specified keyword, displays the topic. If more than one topic matches, displays the Search dialog box with the topics found listed in the Go To list box. If there is no match, displays the Search dialog with the text in the edit box.
HELP_COMMAND	262	Executes a specified help macro.
HELP_SETWINPOS	515	Repositions the help window.
HELP_FORCEFILE	9	Ensures that the specified help file is displayed.

The Help Topic Parameter (dwData)

The *dwData* parameter is used to specify additional information that WinHelp needs to execute the command specified in the fuCommand parameter. The format and data type of the *dwData* parameter depend upon the value of *fuCommand* passed when your application calls **WinHelp**. In the case of HELP_CONTEXT, *dwData* is a long integer specifying the context number of the topic for which the application is requesting Help. For other *fuCommand* values, *dwData* might be a long pointer to a string or special data structure.

The following list describes the format of *dwData* for each value of *fuCommand.*

fuCommand value	dwData format
HELP_CONTEXT	A **LONG** integer containing the context number for the topic. Instead of using HELP_INDEX, HELP_CONTEXT can use the value -1.
HELP_QUIT	Ignored.
HELP_INDEX	Ignored.
HELP_HELPONHELP	Ignored.
HELP_SETINDEX	A **LONG** integer containing the context number for the topic you want to be the index.
HELP_KEY	A long pointer to a string which contains a keyword for the desired topic.
HELP_MULTIKEY	A long pointer to the MULTIKEYHELP structure, as defined in WINHELP.TXT. This structure specifies the table footnote character and the keyword.
HELP_CONTENTS	Same as HELP_INDEX
HELP_SETCONTENTS	Same as HELP_SETINDEX
HELP_CONTEXTPOPUP	An unsigned long integer specifying the context number of the topic to be displayed
HELP_PARTIALKEY	A long pointer to a string to be searched for in the keyword list. To display the Search dialog box without specifying a keyword, pass a long pointer to an empty string.
HELP_COMMAND	A long pointer to a string that contains a valid help macro.
HELP_SETWINPOS	A long pointer to a structure of type HELPWININFO, as described in WINDOWS.H. This structure specifies the size and position of the Help window.
HELP_FORCEFILE	Ignored

Using Help in Applications Written in C

Windows applications can offer help to their users by using the **WinHelp** function to start Windows Help and display topics in the application's Help file. The **WinHelp** function gives a Windows application complete access to the Help file, as well as to the menus and commands of Windows Help. If you haven't already done so, read the section "Explanation of the WinHelp Function" on page 270 for a complete discussion of WinHelp's functionality.

Choosing Help from the Help Menu

Every application should provide a Help menu to allow the user to open the Help file with either the keyboard or the mouse. The Help menu should contain at least one Contents menu item that, when chosen, displays the contents or the main topic in the Help file. To support the Help menu, the application's main window procedure should check for the Contents menu item and call the **WinHelp** function, as in the following example:

```
case WM_COMMAND:
     switch (wParam) {
     case IDM_HELP_CONTENTS:
         WinHelp(hwnd, "myhelp.hlp", HELP_CONTENTS, 0L);
         return 0L;
         .
         .
         .

     }
     break;
```

You can add other menu items to the Help menu for topics containing general information about the application. For example, if your Help file contains a topic that describes how to use the keyboard, you can place a Keyboard menu item on the Help menu. To support additional menu items, your application must specify the context string or context number for the corresponding topic when it calls the **WinHelp** function. The following example, which responds to a command message from the menu identified by the constant IDM_HELP_KEYBOARD, uses a Help macro to specify the context string "UsingTheKeyboard.58" for the Keyboard topic:

```
case IDM_HELP_KEYBOARD:
     WinHelp(hwnd, "myhelp.hlp", HELP_COMMAND,
    (LPSTR)"JumpID(\"myhelp.hlp\",\"UsingTheKeyboard.58\")");
     return 0L;
```

A better way to display a topic is to use a context identifier. To do this, the Help file must assign a unique number to the corresponding context string, in the **[MAP]** section of the project file. For example, the following section assigns the number 58 to the context string UsingTheKeyboard.58:

```
[MAP]
HelpContents.1           1
...
...
UsingTheKeyboard.58      58
...
```

An application can display the Keyboard topic by specifying the context identifier in the call to the **WinHelp** function. In the following example, the identifier IDH_USING_THE_KEYBOARD is defined with the keyboard topic's map number:

```
#define IDH_USING_THE_KEYBOARD 58

 WinHelp(hwnd, "myhelp.hlp", HELP_CONTEXT,
(DWORD)IDH_USING_THE_KEYBOARD);
```

Choosing Help with the Keyboard

An application can enable the user to choose a help topic with the keyboard by intercepting the **F1** key. Intercepting this key lets the user select a menu, menu item, dialog box, message box, or control window and view Help for it with a single keystroke.

To intercept the **F1** key, the application must install a message-filter procedure by using the **SetWindowsHook** function. This allows the application to examine all keystrokes for the application, regardless of which window has the input focus. If the filter procedure detects the **F1** key, it posts a WM_F1DOWN message (application-defined) to the application's main window procedure. The procedure then determines which help topic to display.

The filter procedure should have the following form:

```
int FAR PASCAL FilterFunc(nCode, wParam, lParam)
 int nCode;
 WORD wParam;
 DWORD lParam;
 {
     LPMSG lpmsg = (LPMSG)lParam;

     if ((nCode == MSGF_DIALOGBOX || nCode == MSGF_MENU) &&
             lpmsg->message == WM_KEYDOWN && lpmsg->wParam ==
VK_F1)   {
         PostMessage(hWnd, WM_F1DOWN, nCode, 0L);
     }

     DefHookProc(nCode, wParam, lParam, &lpFilterFunc);

     return 0;
 }
```

The application should install the filter procedure after creating the main window, as shown in the following example:

```
lpProcInstance = MakeProcInstance(FilterFunc, hInstance);
 if (lpProcInstance == NULL)
     return FALSE;

 lpFilterFunc = SetWindowsHook(WH_MSGFILTER, lpProcInstance);
```

Like all callback functions, the filter procedure must be exported by the application.

The filter procedure sends a WM_F1DOWN message only when the **F1** key is pressed in a dialog box, message box, or menu. Many applications also display the Contents topic if no menu, dialog box, or message box is selected when the user presses the **F1** key. In this case, the application should define the **F1** key as an accelerator key that starts Help.

To create an accelerator key, the application's resource-definition file must define an accelerator table, as follows:

```
1 ACCELERATORS
 BEGIN
     VK_F1, IDM_HELP_CONTENTS, VIRTKEY
 END
```

To support the accelerator key, the application must load the accelerator table by using the **LoadAccelerators** function and translate the accelerator keys in the main message loop by using the **TranslateAccelerator** function.

In addition to installing the filter procedure, the application must keep track of which menu, menu item, dialog box, or message box is currently selected. In other words, when the user selects an item, the application must set a global variable indicating the current context. For dialog and message boxes, the application should set the global variable immediately before calling the **DialogBox** or **MessageBox** function. For menus and menu items, the application should set the variable whenever it receives a WM_MENUSELECT message. As long as identifiers for all menu items and controls in an application are unique, an application can use code similar to the following example to monitor menu selections:

```
case WM_MENUSELECT:
     /*
      * Set dwCurrentHelpId to the Help ID of the menu item
that is
      * currently selected.
      */

     if (HIWORD(lParam) == 0)             /* no menu selected
*/
         dwCurrentHelpId = ID_NONE;

     else if (lParam & MF_POPUP) {        /* pop-up selected
*/
         if ((HMENU)wParam == hMenuFile)
             dwCurrentHelpId = ID_FILE;
         else if ((HMENU)wParam == hMenuEdit)
             dwCurrentHelpId = ID_EDIT;
         else if ((HMENU)wParam == hMenuHelp)
             dwCurrentHelpId = ID_HELP;
         else
             dwCurrentHelpId = ID_SYSTEM;
     }

     else                                 /* menu item
selected */
         dwCurrentHelpId = wParam;

     break;
```

In this example, the *hMenuFile*, *hMenuEdit*, and *hMenuHelp* parameters must previously have been set to specify the corresponding menu handles.

An application can use the **GetMenu** and **GetSubMenu** functions to retrieve these handles.

When the main window procedure finally receives a WM_F1DOWN message, it should use the current value of the global variable to display a help topic. The application can also provide Help for individual controls in a dialog box by determining which control has the focus at this point, as shown in the following example:

```
case WM_F1DOWN:
     /*
      * If there is a current Help context, display it.
      */

     if (dwCurrentHelpId != ID_NONE) {
         DWORD dwHelp = dwCurrentHelpId;

         /*
          * Check for context-sensitive Help for individual
dialog
          * box controls.
          */

         if (wParam == MSGF_DIALOGBOX) {
             WORD wID = GetWindowWord(GetFocus(), GWW_ID);
             if (wID != IDOK && wID != IDCANCEL)
                 dwHelp = (DWORD) wID;
         }

         WinHelp(hWnd, szHelpFileName, HELP_CONTEXT, dwHelp);

         /*
          * This call is used to remove the highlighting from
the
          * System menu, if necessary.
          */

         DrawMenuBar(hWnd);
     }

     break;
```

When the application ends, it must remove the filter procedure by using the **UnhookWindowsHook** function and free the procedure instance for the function by using the **FreeProcInstance** function.

Choosing Help with the Mouse

An application can enable the user to choose a help topic with the mouse by intercepting mouse input messages and calling the **WinHelp** function. To distinguish requests to view Help from regular mouse input, the user must press the SHIFT+F1 key combination. In such cases, the application sets a global variable when the user presses the key combination and changes the cursor shape to a question-mark pointer to indicate that the mouse can be used to choose a help topic.

To detect the SHIFT+F1 key combination, an application checks for the VK_F1 virtual-key value in each WM_KEYDOWN message sent to its main window procedure. It also checks for the VK_ESCAPE virtual-key code. The user presses the ESC key to quit Help and restore the mouse to its regular function. The following example checks for these keys:

```
case WM_KEYDOWN:
     if (wParam == VK_F1) {

         /* If Shift-F1, turn Help mode on and set Help
cursor. */

         if (GetKeyState(VK_SHIFT)) {
             bHelp = TRUE;
             SetCursor(hHelpCursor);
             return (DefWindowProc(hwnd, message, wParam,
lParam));
         }

         /* If F1 without shift, call Help main index topic.
*/

         else {
             WinHelp(hwnd,"myhelp.hlp",HELP_CONTENTS,0L);
         }
     }

     else if (wParam == VK_ESCAPE && bHelp) {

         /* Escape during Help mode: turn Help mode off. */

         bHelp = FALSE;
         SetCursor((HCURSOR) GetClassWord(hWnd,
GCW_HCURSOR));
     }

     break;
```

Until the user clicks the mouse or presses the **ESC** key, the application responds to WM_SETCURSOR messages by resetting the cursor to the arrow and question-mark combination.

```
case WM_SETCURSOR:
     /*
      * In Help mode, it is necessary to reset the cursor in
response
      * to every WM_SETCURSOR message. Otherwise, by default,
Windows
      * will reset the cursor to that of the window class.
      */

     if (bHelp) {
         SetCursor(hHelpCursor);
         break;
     }

     return (DefWindowProc(hwnd, message, wParam, lParam));

 case WM_INITMENU:
     if (bHelp) {
         SetCursor(hHelpCursor);
     }

     return (TRUE);
```

If the user clicks the mouse button in a nonclient area of the application window while in Help mode, the application receives a WM_NCLBUTTONDOWN message. By examining the *wParam* value of this message, the application can determine which context identifier to pass to **WinHelp**.

```
case WM_NCLBUTTONDOWN:
     /*
      * If in Help mode (Shift+F1), display context-sensitive
      * Help for nonclient area.
      */

     if (bHelp) {
         dwHelpContextId =
             (wParam == HTCAPTION) ?(DWORD) HELPID_TITLE_BAR:
             (wParam == HTSIZE) ? (DWORD) HELPID_SIZE_BOX:
             (wParam == HTREDUCE) ? (DWORD)
HELPID_MINIMIZE_ICON:
             (wParam == HTZOOM) ? (DWORD)
HELPID_MAXIMIZE_ICON:
             (wParam == HTSYSMENU) ?(DWORD)
HELPID_SYSTEM_MENU:
             (wParam == HTBOTTOM) ? (DWORD)
HELPID_SIZING_BORDER:
             (wParam == HTBOTTOMLEFT) ? (DWORD)
HELPID_SIZING_BORDER:
             (wParam == HTBOTTOMRIGHT) ?(DWORD)
HELPID_SIZING_BORDER:
             (wParam == HTTOP) ?(DWORD) HELPID_SIZING_BORDER:
             (wParam == HTLEFT) ?(DWORD)
HELPID_SIZING_BORDER:
             (wParam == HTRIGHT) ?(DWORD)
HELPID_SIZING_BORDER:
             (wParam == HTTOPLEFT) ?(DWORD)
HELPID_SIZING_BORDER:
             (wParam == HTTOPRIGHT) ? (DWORD)
HELPID_SIZING_BORDER:
             (DWORD) 0L;

         if (!((BOOL) dwHelpContextId))
             return (DefWindowProc(hwnd, message, wParam,
lParam));
         bHelp = FALSE;
         WinHelp(hWnd, szHelpFileName, HELP_CONTEXT,
dwHelpContextId);
         break;
     }

     return (DefWindowProc(hWnd, message, wParam, lParam));
```

If the user clicks a menu item while in Help mode, the application intercepts the WM_COMMAND message and sends the Help request:

```
case WM_COMMAND:

     /* In Help mode (Shift-F1)? */

     if (bHelp) {
         bHelp = FALSE;
         WinHelp(hWnd,szHelpFileName,HELP_CONTEXT,
(DWORD)wParam);
         return NULL;
     }
```

Searching for Help with Keywords

An application can enable the user to search for help topics based on full or partial keywords. This method is similar to employing the Search dialog box in Windows Help to find useful topics. The following example searches for the keyword “Keyboard” and displays the corresponding topic, if found:

```
WinHelp(hwnd, "myhelp.hlp", HELP_KEY, "Keyboard");
```

If the topic is not found, Windows Help displays an error message. If more than one topic has the same keyword, Windows Help displays only the first topic.

An application can give the user more options in a search by specifying partial keywords. When a partial keyword is given, Windows Help usually displays the Search dialog box to allow the user to continue the search or return to the application. However, if there is an exact match and no other topic exists with the given keyword, Windows Help displays the topic. The following example opens the Search dialog box and selects the first keyword in the list starting with the letters *Ke*:

```
WinHelp(hwnd, "myhelp.hlp", HELP_PARTIALKEY, "Ke");
```

When the HELP_KEY and HELP_PARTIALKEY values are specified in the **WinHelp** function, Windows Help searches the K keyword table. This table contains keywords generated by Doc-To-Help from the index entries in your document. An application can search alternative keyword tables by specifying the HELP_MULTIKEY value in the **WinHelp** function. In this case, the application must specify the footnote character and the full keyword in a **MULTIKEYHELP** structure, as follows:

```
HANDLE hmkh;
 MULTIKEYHELP far *mkh;
 char *szKeyword = "Frame";
 WORD wSize;

 wSize = sizeof(MULTIKEYHELP) + lstrlen(szKeyword);

 hmkh = GlobalAlloc(GHND, (DWORD)wSize);
 if (hmkh == NULL)
     break;
 mkh = (MULTIKEYHELP far *) GlobalLock(hmkh);

 mkh->mkSize    = wSize;
 mkh->mkKeylist = 'L';
 lstrcpy(mkh->szKeyphrase, szKeyword);

 WinHelp(hwnd, "myhelp.hlp", HELP_MULTIKEY, (DWORD)mkh);

 GlobalUnlock(hmkh);
 GlobalFree(hmkh);
```

If the keyword is not found, Windows Help displays an error message. If more than one topic has the keyword, Windows Help displays only the first topic. (For a full description of the **MULTIKEYHELP** structure, see the *Microsoft Programmer's Reference*, *Volume 3*.)

Applications cannot use alternative keyword tables unless the **MULTIKEY** option is specified in the **[OPTIONS]** section of the project file.

Displaying Help in a Secondary Window

An application can display help topics in secondary windows instead of in Windows Help's main window. Secondary windows are useful whenever the user does not need the full capabilities of Windows Help. The Windows Help menus and buttons are not available in secondary windows.

To display Help in a secondary window, the application specifies the name of the secondary window along with the name of the Help file. The following example displays the help topic having the context identifier IDM_FILE_SAVE in the secondary window named wnd_menu:

```
WinHelp(hwnd, "myhelp.hlp>>wnd_menu", HELP_CONTEXT,
IDM_FILE_SAVE);
```

The name and characteristics of the secondary window must be defined in the **[WINDOWS]** section of the project file. Doc-To-Help writes these settings for you when you specify windows characteristics through the Doc-To-Help Window Settings dialog box.

Windows Help displays the secondary window with the initial size and position specified in the **[WINDOWS]** section. However, an application can set a new size and position by specifying the HELP_SETWINPOS value in the **WinHelp** function. In this case, the application sets the members in a **HELPWININFO** structure to specify the window size and position. The following examples sets the secondary window wnd_menu to a new size and position:

```
HANDLE hhwi;
 LPHELPWININFO lphwi;
 WORD wSize;
 char *szWndName = "wnd_menu";
wSize = sizeof(HELPWININFO) + lstrlen(szWndName);
 hhwi = GlobalAlloc(GHND, wSize);
 lphwi = (LPHELPWININFO)GlobalLock(hhwi);

 lphwi->wStructSize = wSize;
 lphwi->x           = 256;
 lphwi->y           = 256;
 lphwi->dx          = 767;
 lphwi->dy          = 512;
 lphwi->wMax        = 0;
 lstrcpy(lphwi->rgchMember, szWndName);

 WinHelp(hwnd, "myhelp.hlp", HELP_SETWINPOS, lphwi);

 GlobalUnlock(hhwi);
 GlobalFree(hhwi);
```

Canceling Help

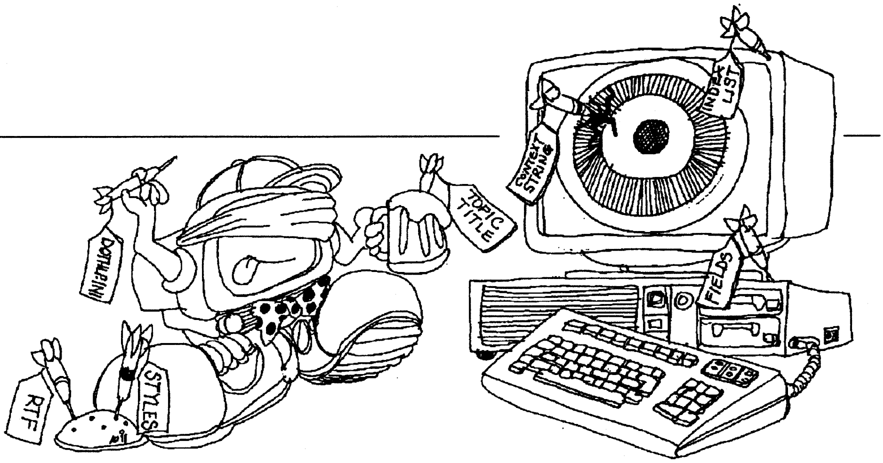
Windows Help requires an application to explicitly cancel Help so that Windows Help can free any resources it used to keep track of the application and its Help files. The application can do this at any time.

An application cancels Windows Help by calling the **WinHelp** function and specifying the HELP_QUIT value, as shown in the following example:

```
WinHelp(hwnd, "myhelp.hlp", HELP_QUIT, NULL);
```

If the application has made any calls to the **WinHelp** function, it must cancel Help before it closes its main window (for example, in response to the WM_DESTROY message in the main window procedure). An application needs to call **WinHelp** only once to cancel Help, no matter how many Help files it has opened. Windows Help remains running until all applications or dynamic-link libraries that have called the **WinHelp** function have canceled Help.

Glossary of Terms



Glossary of Terms

Auxiliary Topic

A topic that can only be accessed from a link created with the Insert Hypertext Link command. An auxiliary topic cannot by accessed from the contents topic, from within the help file's hierarchy, using keyword search or from within a browse sequence.

Bookmark

A name which refers to a location in a document. You use bookmarks to create cross references and to move quickly to remote locations of your document.

Browse Sequence

A browse sequence is simply a series of topics that appear in a sequence when the user chooses the << and >> buttons. Clicking these buttons is much like turning pages in a book.

Button Text

Text that appears as a jump in the Help file. It will be underlined and usually green. Clicking on the button text will jump you to the associated Help topic.

Caption

Text which appears in the margin to provide additional information about something in the adjacent paragraph.

Compiling

The process of converting an RTF file into a Help file, which the Help Engine (WINHELP.EXE) can display as online help.

Context String

A unique name which identifies each help topic, so that it can be accessed by hypertext jumps, keyword search, and context-sensitive help. Doc-To-Help creates these context strings for you.

Context-Sensitive Help

An implementation of online help which automatically brings up a Help screen relevant to the action the user was trying to perform when he or she invoked Help.

Cross Reference

A reference to another location in a document, which appears as text, and which is identified by its bookmark name (e.g., For more information, see "Basic Lawn Mower Repair"). Cross references are created using a Ref field.

Custom Index Entries

Index entries that are relevant to a particular occurrence of an index target.

Default Index Tag

An Index Tag marked for quick placement in the document, using the "Insert Default Entries" button. Index tags that will be placed at most instances of an index target should be marked as Default.

Definition

Text which appears when you click on a word in a Help topic that is marked with a dotted underline.

Discretionary Indexing

The first phase of building the index, in which any index targets marked for "Prompt at Each Instance" are processed. Each occurrence of the index target in the document will be found, and you'll have the option of inserting Index Entries at whichever locations you choose.

DOC2HELP.INI

The Doc-To-Help settings file, which contains information specific to each Doc-To-Help project, including the names of the document files included, contents of the index list and preferences you choose from the Tools Doc-To-Help Options command.

Field

A field is a special set of codes that instructs Word to insert information into a document.

Hanging Indent

A paragraph format in which the first line of the paragraph starts farther to the left than the lines that follow.

HC.EXE

Microsoft Windows 3.0 Help compiler which compiles an RTF file into a 3.0 Help file.

HC31.EXE

Microsoft Windows 3.1 Help compiler which compiles an RTF file into a Windows 3.1 Help file. The Windows 3.1 Help engine (WINHELP.EXE version 3.1) is required to read a Help file compiled by HC31.EXE. (Note the Windows 3.1 Help engine can read both 3.0 and 3.1 Help files.)

HCP.EXE

Protected mode version of the Windows 3.1 Help compiler. Use this compiler if you run out of memory when compiling using HC31.EXE.

Help Compiler

An executable file (HC.EXE, HC31.EXE or HCP.EXE) which compiles an RTF file into a Windows Help file.

Help Project File

A text file that tells the Help compiler where your documents and bitmaps are saved, what your context string mappings are, how you want the help captions to read, and so on. The Help project file has the same name as the RTF file, but with an extension of HPJ.

Help Topic

A screen in a Help file which contains information about a certain topic.

Hypertext Jump

A word or phrase marked with an underline that, when clicked by a user, jumps to another topic.

Index Entry Field

A field which instructs Word for Windows, when updating the Index, to include an entry pointing to the page containing the Index Entry field, or to a range of pages containing a Bookmark specified in the Index Entry field.

Index Tag

The phrase associated with a word in the index target list, as you want it to appear in your document's index.

Index Target

Word or phrase that is searched for when you build the index. Each index target has index tags associated with it; i.e., words or phrases in the index that point to the target location.

Keyword

A word or phrase that points to one or more Help Topics. When you click the Search button in Help, you choose from a list of Keywords, and are then offered a list of Help Topics with which that Keyword is associated.

Non-Scrolling Region

The Windows 3.1 Help Compiler allows you to define a Non-Scrolling Region at the beginning of a help topic. The Non-Scrolling Region remains at the top of the Help Window even when you scroll downwards through the topic. This is typically used to keep a descriptive heading paragraph visible at all times. You can include non-scrolling regions in both the main Help window and in secondary windows.

Page Reference

A reference to another location in a document, which appears as a page number, and which is identified by its bookmark name (e.g., For more information, see page 9). Page references are created using a Pageref field.

Pageref Field

A Word for Windows field which displays the page number of a page that contains the bookmark you are referencing. Pageref Fields are used to create Page References (e.g., For more information, see page 9).

RD Field

RD fields instruct Word for Windows to reference additional documents when updating the index and table of contents.

Ref Field

A Word for Windows field which displays the text of a bookmark, given the name of the bookmark. Ref fields are used to create cross references (e.g. For more information, see *"Basic Lawn Mower Repair."*).

Related Help Topic

A Help Topic which will be accessible by a hypertext jump, marked by underlined text at the bottom of the current Help Topic.

RTF

Rich Text Format. You create online help by preparing a document in this file format, which is then compiled into an .HLP file.

Secondary Window

A secondary window is an independent Help window that has a title bar and scroll bars but no menu bar or buttons. Secondary windows are resizable and moveable, and they remain open until closed by the user or by a help macro, whether or not the main Help window is open.

Source Code

Programming instructions written in a computer language.

Special Bold

Body text which is formatted in bold and in the same font used for headings.

Supertitle

Text which appears above the title of your document on the title page.

TOC field

A field which when updated generates a table of contents based on headings used in your document.

Topic Title

A title which identifies a Help Topic to the user. Topic titles appear where Hypertext jumps are defined, in the Go To list box resulting from a keyword search, and usually at the top of the Help screen.

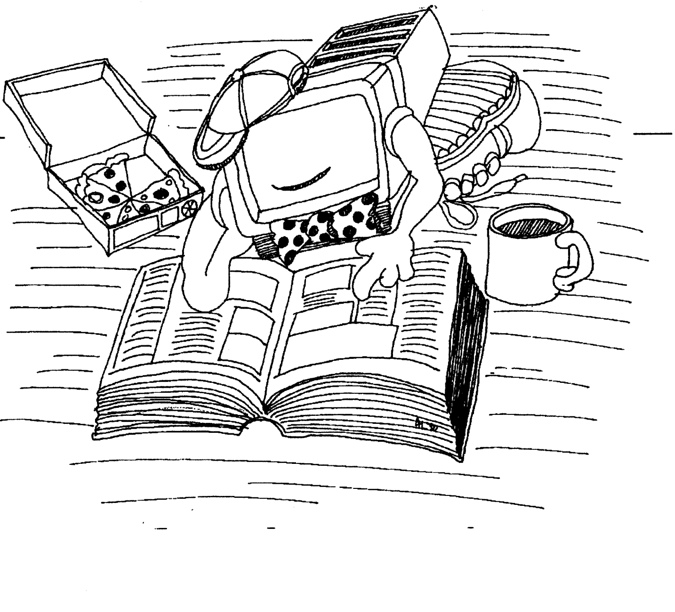
WINHELP.EXE

The Windows Help engine, which you use to view a Help file.

XE Fields

A field which instructs Word for Windows, when updating the Index, to include an entry pointing to the page containing the index tag field, or to a range of pages containing a Bookmark specified in the index tag field.

Index



Index