

Contents



WhizNotes

A Hypertext Organizer and Help Authoring System for Windows
Copyright © 1992-1997, Sanjay Kanade

Help Version 3.S.B(E)

[Your License for Trial Use](#)

[Ordering details](#)

[What you get when you Register](#)

[Magazine quotes on WhizNotes](#)

[Summary of new features](#)

[Web sites](#)

Introduction to WhizNotes:

[Introduction](#)

[The philosophy of WhizNotes](#)

[Getting started](#)

[Quick Start Tutorial](#)

[Startup Tips and Tricks](#)

[Please check the program's speed](#)

[Who needs to create Help files?](#)

[Introduction to Help Authoring with WhizNotes](#)

Using WhizNotes as a Hypertext Organizer

[Manipulating WhizFolder files](#)

[Manipulating topics within a WhizFolder file](#)

[Making use of Windowing features of WhizNotes](#)

[Printing WhizFolder files and topics](#)

[Export/import features of WhizNotes](#)

Using WhizNotes as a Help Authoring System

[Introduction to Help Authoring with WhizNotes](#)

[Advantages of using WhizNotes for help generation](#)

[Basics of Help generation](#)

[HelpHikes Markup Language \(HHML\)](#)

[The magic of Hypertext links](#)

[Bitmaps, hotspots, and Hypergraphics](#)

[Character/paragraph properties](#)

[Help Macros](#)

[Programmer's corner](#)

[The power to customize](#)

Appendix:

[Magazine quotes on WhizNotes](#)

[Limits set in WhizNotes](#)

[Less often used INI flags](#)

[How to uninstall/move WhizNotes](#)

[Please check the program's speed](#)

Temporary files created by WhizNotes
Other programs in the WhizFolder family

What's new

Recent changes since the last shareware release:

- The WhizMenu tool has been made available in the shareware version. See help on the keyword "WhizMenu." If you are a help authoring user, just right-click on a topic window to see and use a WhizMenu to insert HHML statements. If you are a general user, please read the above help topic to see how WhizMenu can be useful for you.
- [Links to other types of documents](#) are also possible with the JumpFile statement!
- A maximized program window used to restore whenever a new folder was opened. This has been corrected.
- On Windows 3.1, if the program was started by double-clicking on a WZN file from the File Manager, a subsequent save of the file produced strange errors leading to a General Protection Fault. This has been fixed.
- Double-click select of a line in the editor has been fixed. Earlier strange characters appeared when you deleted a line selected by the above operation.
- Now, for long help compilations, if you forgot to check the option "No wait for compilation" on the "Help Options" dialog, you can still press "Stop waiting" with the same effect.
- New menu items are available. For more details, highlight those menu items and press F1.
 - "Locate GlobalHelpOptions Definition" on the Tools menu. In addition, the right-click speed menu on the topic list has changed to include these items.
 - "Always keep on top" on the Options menu.
 - "Browse next topic" and "Browse previous topic" on the File menu when a topic window is open.
 - "Save selected topics to text file" on the File menu.

Changes since WhizNotes 2.13/2.14

There are now three programs in the WhizFolder family. The other two programs are WhizFolders and HelpHikes Pro. For more details, please see [Other programs in the WhizFolder family](#).

Lots of changes have occurred since WhizNotes 2.13/2.14. I am not going to list the numerous small improvements and fixes. Only the major improvements are listed here.

General new features:

- The program has been highly optimized for speed. Although it remains a 16 bit program, you will find a terrific speed improvement in all the operations. Please don't be misled into thinking that 32 bit programs are faster than 16 bit programs. A highly optimized 16 bit program can be faster than a poorly programmed 32 bit program.
- The program can run on both Windows 3.1 and Windows 95 systems. On Windows 95, it detects the presence of long name file system and allows you to save folder files by long file names. You will also find nice tips to create desktop shortcuts to your folders in this help.
- A Spelling-Checker is now available as an add-on program for the registered version only. Please see the Ensign Systems web site (<http://www.ensignpub.com/>) to see what all

languages can be ordered for the Spelling-Checker. If you don't find the language of your choice, please send an email message to the author at skanade@superb.net. If enough users are ready to pay for a new language, it can be made available.

- Online help has been revamped. On Windows 95, you will see the collapsible Contents topic dialog box and you will also notice a hierarchical keyword index.
- Now you have an option to restore the complete desktop. This restores all the WhizFolders, topics, and text files which were open at the time of last exit of the program. Use this option on General Desktop Options. **(only available in the registered version)**
- When copying the topics to other folders by drag & drop, you can press Shift key for a move operation.
- In addition to drag & drop, there are also other convenient ways to paste a jump statement via the clipboard. Just right-click on the topic list and you can copy a jump statement for the selected topics. This can also be done on an open topic window on the edit menu.
- You can even drag and insert a jump between topics of different folders. This inserts a JUMPWZN statement which when double-clicked can even open the folder. [Links to other types of documents](#) are also possible with the JumpWzn statement!
- New powerful tool WhizMenu is available to paste your own canned text or to enter HTML statements. This menu appears when you right-click on an editor. On Windows 95, shift-right-click will get you the original edit menu for Windows 95. Sample HTML (web authoring) coding examples are also given in another WhizMenu file. **(only available in the registered version)**
- You can right click outside the topic list to see the list status.
- You can set up print margins and various options for printing from the menu "Options-Printing options".
- Now you can press Shift-F2 (menu Search - Locate self...) on a topic window to quickly locate itself in its parent WhizFolder. The parent comes on top with that topic selected in the list. Very useful!
- Editor now correctly handles files approaching 32K in size. Also, now Ctrl-Y can be used to delete a line and double-click on the beginning of a line selects the line.
- New **Desktop Options for the WhizFolder** windows:
 - New topic prefix of Date/Time can be specified so that your WhizFolder works like a journal.
 - "Don't open topic window on add topic" to quickly build a list of new topics.
 - "Don't create backup file" to save space.
- New **General Desktop Options**:
 - Restore complete desktop on startup. **(only available in the registered version)**
 - No confirm for re-order topics. If you have to rearrange your topic list too frequently with drag & drop, use this.
 - No confirm for move to another WhizFolder. If you have to frequently move topics

between WhizFolder files with drag & drop, use this.

- Many **new menu operations** are available. I will just list them here. To see help on them, just select them and press F1.

General new items:

- Tools - Check Spelling
- Tools - Forget "ignore all", "change all" each time
- Options - Printing options...
- Options - Spelling-Checker options...

For topic windows:

- File - Save WhizFolder as...
- File - Save all
- Edit - Copy special... topic names, topic contents, jump statements
- Edit - Convert into paragraph.
- Edit - Convert to lower/upper case
- Search - Locate self in parent list

For folder windows:

- File - Print list of topics
- Edit - Copy topic names, jump statements
- Edit - Enlarge topic list font, reduce topic list font
- Edit - Enable horizontal scrolling for the topic list
- Edit - Force word-wrap for selected topics
- Edit - Show topic list status
- Search - Select topics containing text

New features for help generation:

In the evaluation version, "Export non-help RTF" and GlobalHelpOptions has been enabled. Some advanced help generation features like Browsing sequences and help macros are no longer available in the evaluation version to reduce the support cost for non-registered users.

Extensive changes have occurred since WhizNotes 2.13/2.14. Only the major points will be listed here. It's best to look at the Contents of the online help which has been reorganized. You will find lots of new topics which will interest you.

Also note that now another more advanced help authoring system is available called HelpHikes Pro. For details, please see [HelpHikes Pro, additional features](#).

- The smart template insertion tool [WhizMenu](#) has been made available in the shareware version.
- The statements like [~jump...] and [~Bold...] are now called HHML statements. See [What is HHML?](#) New WhizNotes adds many many more statements to HHML set. For a complete list, see [HHML: List of statements supported by WhizNotes](#).
- Now the new HCW compiler is also supported. See [Setting up the Help Compilation environment](#).
- New powerful tool WhizMenu is available to quickly enter HHML statements. This menu appears when you right-click on an editor. On Windows 95, shift-right-click will get you the original edit menu for Windows 95. See [WhizMenu for quick entry of canned text and](#)

HHML statements. **(only available in the registered version)**

- Advantages of using WhizNotes for help generation are explained.
- The RTF and HPJ files are not deleted after the compilation.
- There are also other convenient ways to paste a jump or ref statement via the clipboard. Just right-click on the topic list and you can copy a jump or ref statement for the selected topics. This can also be done on an open topic window on the edit menu.
- More details on where to get the help compiler are included.

Here's a list of topics. They have been marked with "New" or "Additions" for your convenience.

- Additions: Terminology of help generation
- New: All kinds of Jumps & References to another topic.
- New: Defining keywords for topics.
- Additions: Keywords, macros, and other characteristics of topics
- New: How to insert bitmaps into your help files
- New: Bitmaps as hotspots
- Additions: Making your text bold, italic, etc..
- New: Starting a new line within a paragraph
- New: Displaying text in tables
- New: Features for context sensitive help
- New: Hints on testing the help file
- New: Modifying RTF and HPJ files with another tool
- New: The power of the HelpTopic paradigm
- Additions: Inserting direct RTF code
- Additions: Global options to control help generation in general
- New: Customizing Help Project (HPJ) file

Also see:

Other programs in the WhizFolder family

Introduction to Help Authoring with WhizNotes

See a demo of Help generation:

Before you begin, why not see a demo! If you think, you have a help compiler installed on your system, just open the sample folder **SAMPLHLP.WZN** with the menu **Open-WhizFolder..** Then, click on the menu **Tools-Create Help** to see WhizNotes in action. You'll see the topics in the sample folder converted to a Help file in a matter of seconds. If the help compiler is not installed properly, you'll get a suitable error message. In that case, please consult the topic [Setting up the Help Compilation environment](#).

Now that you've seen a demo, open a few topics in the sample folder by double-clicking on them. They're quite easy to understand, and if something is not clear, you can always search the online help for more details. I'm sure you'd like to make a few changes in the text and try generating help again. By all means, do it! You can even try [inserting more hypertext jumps](#) between topics. It's as easy as a drag & drop. Please see [More about the demo](#) for more details on this demo WZN file.

You're beginning to get an idea of how easy it is to create hypertext help with WhizNotes. What you've seen are only a few of the basic help capabilities. Want to know complete details on what WhizNotes has to offer for help generation? Please read [Help generation with WhizNotes](#).

Introduction to Help Authoring with WhizNotes:

WhizNotes is a Help Authoring System. Creating help with it is as easy as adding topics to a folder, typing text in separate topic windows, linking the topics with a simple drag and drop of topic names and then pushing a button to compile the help file. In fact, creating help is so easy that you will be tempted to make help files even for your regular everyday notes. WhizNotes also supports advanced features for customizing your help RTF generation with an innovative customizable macro language called [HHML](#). You can also treat the topics as objects and specify keywords, browsing sequences, help macros to those objects conveniently, in free-format, all in one place.

Software requirements:

WhizNotes can run on Windows 3.1, Windows 95, or later. For authoring Windows Help files with WhizNotes, you'll need to additionally install Microsoft help compiler 3.1, or later. For more details on compilers, please see [Setting up the Help Compilation environment](#).

Get "HelpHikes Pro" for more advanced features:

If you are really into help authoring as a professional, you should get the more advanced Help Authoring program, [HelpHikes Pro](#) which also supports all the new features available in Windows 95 help. For details, please see [HelpHikes Pro, additional features](#).

Introduction

- Do you often feel overwhelmed with the bulk of information that you have to deal with?
- Do you constantly come up with new ideas, but have trouble recalling them later?
- Do you find it difficult to reorganize your thoughts when you decide to write up the first draft of a document?
- Do you wish that you could create your very own electronic reference notes on your favorite subjects, with hypertext links joining these notes?
- Have you always wondered how Windows help files are created with all those nice hypertext jumps to other topics? Do you wish that you could convert your notes into Windows hypertext help?

Well, don't blame yourself for any of these situations as they are indeed the manifestations of your creative mind. Do not lose those precious thoughts away! Give your imaginative mind the proper writing tool that it deserves. Give it WhizNotes and see your productivity soar as never before!

Welcome to the world of WhizNotes:

Are you still looking for the ideal note-taking program? A program that allows you to take random notes and also helps you organize the vast amount of random information that you collect every day. A program where you can find your notes on any subject easily, a program that allows you to keep large bodies of text, related or unrelated, grouped into folders of your choice. Well, look no more. You have arrived! WhizNotes can do all that for you.

What's more, WhizNotes even lets you link your notes by hypertext links by a simple drag and drop. The advantage of hypertext is that you can jump from a note to another note by double-clicking on these links. In WhizNotes' hypertext scheme, any note which is not already open is opened for you and if the note is contained in another folder, even that folder is opened for you. Combine this with the ability to give long names to the notes and the folders, and you will be delighted with how much you can accomplish with WhizNotes.

To top it all, WhizNotes is also a [Help Authoring System](#)! If you have the Microsoft Help Compiler, you are just a mouse-click away from converting your hypertext notes into Windows hypertext help. Whatever links you created between the notes are automatically converted into Windows hypertext jumps. Want to see a demo? See [Introduction to Help Authoring with WhizNotes](#).

Start using WhizNotes and say goodbye to your journals, logs and diaries. No restrictions. You can enter free-format text in your notes and search for it in free-format as well.

Once you start using WhizNotes, you might wonder how you ever managed without this versatile tool. After knowing more about WhizNotes, you will find that it does not fit into any single category of programs that you have known so far. It may look like a PIM (Personal Information Manager) at first glance. But, it is also a research tool, an organizer, a study aid, a thinking aid, and so on. Its variety of uses are only limited by your imagination.

WhizNotes' information management facility is a great help in making notes of any kind, whether you are a student, a writer, a programmer or anyone with a need to get more control on your random information.

How it helps in writing:

Please don't be misled into thinking that WhizNotes is only a random information manager. It has its uses for organized information as well. It can help you in your writing. How?

Let us take an example of a document preparation process that is starting from scratch. We all know how painful it can be to prepare that *first draft* of a document. You may have all the ideas on how to go about writing that document but all your thoughts seem tangled to begin with. WhizNotes helps you overcome this inertia effectively and easily. How? It lets you classify your information into separate topics instantly. WhizNotes provides powerful topic management facilities. Therefore, you can concentrate on the subject matter of the document without worrying about its structure.

The philosophy of WhizNotes:

Want to know more about the concepts behind WhizNotes? Why not take a look at [The philosophy of WhizNotes](#).

To mention a few of the more important features, WhizNotes gives you an ability to:

- Put your notes in one or more topics.
- Choose any descriptive topic name, even with embedded blanks and special characters.
- Group all these topics together in a WhizFolder.

This way, you can create as many groups as you want by simply using different WhizFolder files. WhizNotes fully supports Windows 95 long names so that you get the long naming facility for your folders too. On Windows 3.1, you can use a pseudo long name facility for a WhizFolder which appears as the title of the WhizFolder window for easy recognition.

To get a better idea of the grouping flexibility provided by WhizNotes, please take a look at the [Examples of WhizFolder files](#).

- Create hypertext links between the related notes so that you can jump from note to note and note to a note in another folder.
- With just a click on the menu, generate Windows hypertext help, provided you have the Microsoft help compiler.
- Print a neat listing of one or more topics including a table of contents.

Start using it immediately:

Want to start using WhizNotes right away? You can do that and take a look at the online help as you go. Remember, you can get help on any menu item by selecting it and pressing the key F1.

Take a look at [Getting started](#) to see how to quickly create a new folder and add topics to it.

Get the latest information:

Get the latest information on WhizNotes. Visit the Ensign System's web page at:

<http://www.ensignpub.com/>

You might find more documentation, tips and FAQs on the program based on the latest questions asked by the users.

Also see:

[Other programs in the WhizFolder family](#)

If you are interested in the use of WhizNotes as a Help Authoring System, you may as well see the topic [Compile the sample help folder](#).

Creating a new folder:

- Click on the menu **File-New WhizFolder** to create a new folder.
- Press Insert key, and enter a topic name for a new topic.
- Enter text for your topic in the topic window that is opened for you.
- Add more topics in the same way.
- Click on the menu **File-Save WhizFolder** and type a file name for your folder.
- Click on the menu **File-Close WhizFolder**. This will close the folder along with its children topics.

Reopening the folder and modifying text:

- Click on the File menu and you will see the name of the last saved folder at the bottom. Click on it to reopen the folder.
- You will see the list of topics that you added last time. Double-click on any topic to open it.
- Modify its text. Modify text for other topics as needed.
- Click on the menu **File-Save WhizFolder**. This will save all modifications.

You have just learnt how to create a folder and enter or modify topics in it. Simple, isn't it? This much can get you started to enter your notes right away. But remember, there are many other features which you should know of. So, please look at the contents of the online help when you have some time. Also take a look at the [Quick Start Tutorial](#) to see examples of some other operations.

WhizNotes as a Help Authoring System:

If you are interested in the use of WhizNotes as a Help Authoring System, you should look at the topic [Compile the sample help folder](#).

Restoring the desktop of a folder:

You can also put on 'Restore desktop' option for the WhizFolder by menu **Options-Desktop options for WhizFolder**. If you do that, any open topics at the time of last close will be restored when you open a WhizFolder. Try it. You will like it.

In the registered version, there's also a feature to restore the entire desktop including all the WhizFolders and the Text Files at the time of last closing.

The philosophy of WhizNotes

When you start writing, what seems to be more important to you: the text content of your document, or its appearance?

You will agree that the *text content* is more important in the initial research phase of a writing project. After all, the *text content* is the soul of a document. If it is not thought out properly then no matter what you do with the document's appearance, it would not be effective.

When you begin to write, your foremost aim is to put your creative thoughts into words. You are not worried about the appearance of the document at this point. Rather, you want an uninterrupted flow of thoughts to your text.

This is also true when you are learning a new subject and making notes, planning something, or just involved in creative writing or thinking.

Creativity and randomness:

Preparing a document from scratch is a creative process. You must have noticed that these thoughts occur to you in a random manner.

For instance, when you are working on one part of the document, you may suddenly get an idea about the same or an altogether different subject. Also, while studying a particular topic, in a flash, you may solve an earlier problem that had been nagging at the back of your mind. Well, the ways of a human mind are mysterious. It seems to work in the background, that too on multiple problems at a time. How can you cope with this train of random ideas effectively?

First of all, it is important that you are able to record these ideas as and when they occur. That is not enough! You should also be able to group, identify, retrieve and print your random notes easily. Most of us resort to bits and pieces of paper when dealing with such problems. Isn't it an era to expect more from your computer?

Do word processors really help you in this creative phase?

You will notice that most of the modern writing tools and sophisticated word processors tend to concentrate on only one aspect of documentation: its appearance and appearance only.

A number of excellent word processors are available today. But, unfortunately, there is not much help around when it comes to preparing the document's soul: its subject matter.

When writing, it helps immensely if you are able to write in multiple windows which are open side by side. Each of your thoughts is initially a separate entity and it should get an independent view of its text. This is not possible in normal word processors or text editors. Even if some word processors support outlining, the outlined notes don't get a separate window on the screen. When you become used to the multiple window authoring of WhizNotes, you'd hate to go back to writing a large document in a single window.

The idea that WhizNotes encourages is to divide your writing in pieces, keeping many of these pieces open for writing in separate windows. Finally, organize them properly in one or more folders, then export them in a single document. Later, you can give finishing touches to the text presentation of this document with your word processor.

A word on outlining:

Although WhizNotes can also be used for single-level outlining, its primary use is different and precedes that of an outliner.

There is a fine line between the research and the outlining phases of your writing project. The research phase belongs in your creative mind, whereas outlining is more of a rational process where you have already selected what ideas belong and what don't belong, and you are trying to prioritize them in a sequence and in a hierarchy.

According to many creative writing authorities, you should not engage in outlining too soon, otherwise your first draft is going to suffer. Instead, you should give equal importance to all your random ideas occurring in the research phase with no sequence assigned to the notes whatsoever. In WhizNotes terms, you will be placing each of these ideas in a different topic and will keep the topics in the sorted order. As your research progresses, with time to nurture them--many of these ideas will take a better shape in your creative mind, and you will also gather some new ones. This is the phase where WhizNotes is extremely useful, and there is no other program quite like it.

Using WhizNotes to author a help system:

If you think, help files are only created by programmers for providing online help, you are severely limiting your point of view. See [Who needs to create Help files?](#)

When you are writing a help system, you will feel an even greater need for creating information in pieces and by topics. Also, you will constantly need to establish jumps between your topics and to move from topic to topic to verify the need for the explicit relationship that you created. WhizNotes makes it incredibly simple to do that by allowing you to drag and drop topic names on top of other topics. This inserts jump statements which allow you to move from topic to topic by double-clicking on them. Later when you create the Windows help, these statements are converted into Windows hypertext jumps. The ability to create information in pieces and reorganize it in pieces is vital for any help authoring. Many so called help authoring tools fail to simplify this most important aspect of help writing. They are using a different paradigm; they require you to remember and pre-plan lots of information about your topics, and are often used in the later stages when you have completed the most painful groundwork for your help writing--planning your topic contents and the relationship between them. WhizNotes helps you right from the early stages of the help development: the stages that require a major and most painful chunk of your total effort.

Graphical User Interface

What is auto-indent?

if **auto-indent** is **ON** and you press **Enter** at the end of a line to go to the next line, the caret is positioned on the next line such that it is aligned with the first non-blank character on the preceding line. This is called auto-indent as it indents the next line at the same level as the preceding line. It is very convenient for any type of text editing. Hence, you are advised to always leave auto-indent **ON**.

What is a topic?

A topic is the smallest unit of information that you want to classify and keep as an individual entity. In other words, it is a note that you want to identify by a distinct name. WhizNotes allows you to keep related topics in a single WhizFolder file where they can be easily identified and accessed.

WhizNotes does not restrict you in any way to choose the names for your topics. Topic names can be as long as 255 characters and can contain embedded spaces and special characters. So, even a descriptive phrase can be a topic name. Once you get this flexibility to choose the names that you want, you will never want to be without it.

Also see:

[What is a WhizFolder file?](#)

[Creating a new topic](#)

[Opening a topic for editing](#)

[Renaming a topic](#)

[Deleting a topic](#)

[Operations on multiple topics](#)

What is a WhizFolder file?

A WhizFolder file is a collection of many topics. You create such files with the help of WhizNotes. These WhizFolder files have the extension **.WZN**.

It is up to you to group any topics you want into one **WZN** or WhizFolder file. This way you don't have to remember each of your topic names. All you need to remember is the WhizFolder file name that serves as a folder name.

Also see:

[What is a topic?](#)

[Examples of WhizFolder files](#)

[Creating a new WhizFolder file](#)

Opening an existing WhizFolder file

To open an existing WhizFolder file, select the menu **File-Open WhizFolder...** A standard file-open dialog comes up where you can select the desired WhizFolder file to open.

Once you select a file and press OK, you see that a gray WhizFolder window appears that lists the topics contained in this file.

You can open, delete and add topics to this WhizFolder file.

Please note that WhizNotes allows you to keep multiple WhizFolder file windows open at the same time. You can even drag topics from one WhizFolder file to another.

Also see:

[What is a WhizFolder file?](#)
[Opening from File Manager](#)

Saving a WhizFolder file

To save a WhizFolder file, select the menu **File-Save WhizFolder**. If you have topic windows open and have made changes to them, you can remain assured that all the changes will be recorded in the WhizFolder file.

If the WhizFolder window is **Untitled**, a standard **File-save dialog** appears that allows you to choose a name and a location for the WhizFolder file.

Please note that all the attributes of the topics, for example, their word wrap and auto-indent attributes, are remembered in the WhizFolder file along with the topics. This is indeed very convenient, so that once you create a topic with a certain attribute, it stays that way. Also, the "desktop options for the WhizFolder" are also saved to the WZN file during this operation.

Please note that a save operation is not actually performed if you have made no changes to the WhizFolder since the last save. It is considered a change if any of the topic contents, topic name, or the order has changed. The change in options is not considered a change as they are not so important. For details, please see [Desktop options for a WhizFolder](#).

You can also use **Save all** to save all the open windows, including WZN and text file windows. The save takes place only for those windows which have been modified since the last save operation.

Also see:

[Save WhizFolder as.. menu option](#)

Opening from File Manager

WhizNotes automatically makes an association for **WZN** files in the **File Manager** or **Explorer** for you. So, if you double-click on a **WZN** file, it is opened for you by WhizNotes. Even if WhizNotes is already running, it opens the file for you.

You can also associate extensions for other text files with WhizNotes. The above facility then extends to the text files as well. However, currently WhizNotes can not open text files larger than 32K bytes.

Also see:

[Creating shortcuts to open WZN files](#)
[Startup Tips and Tricks](#)

Save WhizFolder as.. menu option

Use this option when you want to save a WhizFolder file by some other name.

Also see:

[Creating a new WhizFolder file](#)
[Saving a WhizFolder file](#)

Edit menu copy operations for a WhizFolder window

You can copy the contents or names of selected topics to the clipboard using the edit menu operations: **Edit-Copy topic contents** or **Edit-Copy topic names**.

Two other options also allow you to copy [~JUMP ...] and [~REF...] statements for the selected topics to the clipboard. Then you can paste them in another topic at a suitable place to provide a hypertext link to the original topic. This copy operation is also available from the topic list on a right-click menu.

Opening a topic for editing

To open a topic, you have to first select the topic in the WhizFolder window. Then you have three choices available to open a topic for editing:

- **Double-click** on the topic name.
- Or, press **Enter** key.
- Or, select the menu **File-Open Topic...**

You can open more than one topic with the last two methods, provided you have selected them. For details on how to select multiple topics, please see [Operations on multiple topics](#).

If you look at the WhizFolder window, you can recognize the open topics by a red pen that appears on the topic names. Opening an already open topic brings its topic window on top.

You can always close all the topics by selecting the menu **Window-Close All Topics**. Similarly, you can also minimize all the topics to organize your desktop.

Renaming a topic

To rename a topic, just select the topic in the WhizFolder window then select the menu **File-Rename Topic...** A dialog box appears that allows you to change the name of the topic.

Rename operation is also available from an open topic window.

Ordering your topics in the topic list

This feature allows you to customize the order of the topics appearing in the topic list. The default behavior is to view the list of topics sorted alphabetically. But you can uncheck the menu item **Edit-Sorted list of topics** and then rearrange the topics in a desired sequence by moving one or more topics to the top, up, down, or bottom. Using a mouse, you can even drag and drop topics to reorder them. Normally, when you do that a dialog appears to confirm the move. You can put this confirmation off by the dialog that appears on menu **Options-General desktop options**.

To move your topics, use either of the following methods:

- The Edit menu commands to move the topics.
- The keyboard shortcuts for these commands as mentioned on the above menu.
- The arrow buttons on the right which represent move to Top, Up, Down, and Bottom respectively.
- **Drag and drop** the selected topics using the mouse. After the move, the selected topics occupy the position pointed to by the drop. If the target topic is not visible, you can drag the mouse cursor above or below the list to scroll it.

Searching for a topic

You can search for a topic in two ways:

- If you remember one or more beginning characters of a topic name, you can select the menu **Search-Select Topic Beginning With...**. A dialog box appears that allows you to select a topic by typing the beginning characters. For more details see [Select Topic Dialog Box](#).
- If you remember an embedded word or phrase within a topic name or within its contents, you can search for it by selecting menu **Search-Find...**

There's also a [Search and select operation](#) operation available to select a set of topics containing the search text in topic name or the topic contents.

Also see:

[Search and select operation](#)
[Search/replace operations](#)

Search/replace operations

Search operation:

You can use menu **Search-Find...** to search for text in the topic names and their contents. In the search dialog, if you specify **Also Search Topic Contents** then the search goes through each topic name and the topic contents. If a topic contains the search text, the topic window is opened for you showing the text found. If you continue the search by pressing **F3** (menu **Search-Find Next**) then it continues through the rest of the topics in the WhizFolder file.

Note that you can press ESC to cancel a long search operation.

Replace operation:

If you are using an unsorted list of topics, you are allowed to perform a global replace through the topics by selecting the menu item **Search-Replace....** If you are doing a replace through topics with no prompt and for all occurrences then depending on your search specification, it may be a long operation. A **Cancel replace dialog** is provided that allows you to cancel this operation. *It is suggested that you should perform a "Save WhizFolder" just before starting such a global replace. This way if you make a mistake, you can discard all the changes by closing the window and can reopen the file to start again.*

Also see:

[Search and select operation](#)

Inserting a jump to another topic

When you drag a topic from a WhizFolder window onto one of the open topic windows, a dialog box appears that allows you to insert a WhizNotes Jump. Using this jump, you can literally jump to the indicated topic by double-clicking on it.

You can also drag and drop multiple topics. Each jump statement is inserted on a new line, preserving the indentation if any. You can also undo this drop if you change your mind.

If, instead, you want to copy and paste a jump statement, that method is also available from an open topic window (menu **Edit-Copy special...**) or from the topic list on a right-click.

Jumps to other WhizFolders

You can also insert jumps to the a topic in another WhizFolder. Such jumps are inserted as **JumpWzn** statements. These jump statements are of the form [**~jumpwzn wznfile::topic**] where wznfile is full file name for the WZN file. Normally, a double-click on such a statement opens the folder if it's not already open, and then the topic is opened if it's not already open by restore desktop feature of the folder. You can edit these statements to remove the '::

Jumps to external documents!

You can drag & drop any types of files from the Explorer or File Manager to an open topic window. This inserts a JUMPFIL statement which allows you to open that document in its native application. This allows you to attach useful notes to a group of external documents and launch them from within your topics. This way, your notes act as a sort of desktop launchers for those set of files. On Windows 95, you can even point to file system folders and have them open up for you by double-clicking on the JUMPFIL statement.

Also see:

[How jumps are printed?](#)

Operations on multiple topics

WhizNotes allows some operations on multiple topics. For example, you can delete a number of topics at once by selecting them all and performing the delete operation.

How to select multiple topics:

To select multiple topics use standard Windows guidelines. For instance, you can select multiple topics by pressing the control key along with a mouse click. You can also select a range of topics by selecting the first topic in the range and then moving to the other end of the range and pressing the SHIFT key with the mouse click. For more details on how to select multiple topics in a list box, please refer to your Windows documentation.

Select multiple topics on a search criteria:

If you want to select topics based on some text present in their names or contents then use the menu **Search-Select topics containing text**.

WhizNotes currently allows the following operations on multiple selected topics:

- Open topics.
- Delete topics.
- Move or rearrange topics.
- Print topics.
- Copy or Move topics to another WhizFolder by dragging them with the mouse.
- Export topics to a text file.
- Copy names or contents of the selected topics to the clipboard.
- Copy JUMP or REF statements for the selected topics. This is available on the right-click as well as on edit menu's **Copy Special...** operation.
- Export topics as non-help Rich Text Format.
- Force word-wrap on topics.
- Give word count for topics.
- Reset selected topics to default sizes.

Opening multiple WhizFolder and topic windows

WhizNotes uses Windows [Multiple Document Interface](#) that gives you many features for managing your open windows, whether they are WhizFolder file windows or topic windows.

You can go on clicking on topics to open them, and you can go on opening other WhizFolder or text files. The **Window** menu gives you many features by which you can move between the windows, can iconify, tile, or cascade them, or even close all the windows, or only the topic windows. For details, see [Multiple Document Interface](#).

You can also drag one or more topics from one WhizFolder window to another by using the mouse. See [Copying topics from one WhizFolder to another](#)

Manipulating external text files

After using topic management facilities of WhizNotes, you won't go back to managing your text in separate text files on your disk. However, WhizNotes also allows you to view or edit external text files. This helps you exchange text with them using the standard clipboard operations.

- You can create a new text file by selecting menu item **File-Text File...-New Text File**. Then you can save it by any name of your choice by selecting **File-Save As...**
- You can open any text file by selecting **File-Text File...-Open Text File...**. You will notice that the File menu remembers the last 8 WhizFolder or Text files that you visited. You can just click on one of those file names to reopen them quickly.
- Similarly, you can also save one of your topics to an external text file. See [Saving topic to an external text file](#).
- Currently, WhizNotes only allows you to view/edit text files of up to 32K characters in size.
- Word wrap and auto-indenting is also available when editing external text files. However, word wrap can not be remembered within a text file. Rather, you have to remember which file is to be viewed with word wrap option. This is similar to Windows **Notepad**.

Also see:

[Importing topics from a text file](#)

[Exporting selected topics to a text file](#)

[Importing whole text files as topics](#)

Saving topic to an external text file

To save a topic to an external text file, open the topic window by double-clicking on the topic. Then select the menu **Save to text file...**

Special saving for word wrapped topics:

When you select this operation, you are given a choice to break the paragraphs into lines as seen on the topic window. This can be useful when preparing text files for the editors which don't have a word wrap feature.

If you want to export a topic with **word wrapping on a specific column**, you should use the **Export feature** instead of a save.

Copying topics from one WhizFolder to another

If you have one or more WhizFolder windows open, and you want to copy or move one or more topics from one WhizFolder to another, do the following:

- Bring both windows into view. For better visibility, you can minimize the unneeded windows and use menu **Window-Tile** for better view of the desired windows.
- Select the desired topics in one window. For details on how to select multiple topics, please see [Operations on multiple topics](#).
- Then, while keeping the left mouse button pressed, drag the mouse cursor onto the other WhizFolder window and release it.

You will see that the topics are copied to the other WhizFolder file.

To perform a move operation instead of a copy, press the SHIFT key when dragging. After the copy, a dialog appears to confirm the deletion of the moved topics from the original WhizFolder. You can put this confirmation off by the dialog that appears on menu **Options-General desktop options**.

This dragging feature is indeed very convenient:

For instance, you can make one of your WhizFolder files a wastebasket or an archive. You can drag and drop old topics onto this basket and then delete them from the original WhizFolder.

Printing from a topic window

This is the feature that you are likely to use more often when printing a single topic, say a letter. No special headers or footers are printed except for the page numbers. This is what you usually want when printing a single topic.

To print from a topic window, open the topic window by double-clicking on it. Then select the menu **File-Print**. A standard print dialog appears that allows you to select the standard printing options or to set up a printer.

Also see:

[How jumps are printed](#)

[Printing one or more selected topics](#)

[Setting up print margins and other printing options](#)

Printing one or more selected topics

If you select the menu **File-Print selected topics...**, the following actions take place, depending on whether you have selected a single or multiple topics in the WhizFolder window.

Printing a single selected topic:

The topic is printed, and proper headers and footers are inserted by WhizNotes when printing.

Printing multiple selected topics:

You are asked for a title for your printout, then the topics are printed, and proper headers and footers are inserted by WhizNotes. In addition, a table of contents is printed at the end. For details on how to select multiple topics, please see [Operations on multiple topics](#).

You can also print all the topics by selecting the menu **File-Print All Topics**.

[Print list of topics](#) is also available on the File menu.

Also see:

[How jumps are printed](#)

[Printing from a topic window](#)

[Setting up print margins and other printing options](#)

Using Quick draft printing for convenience

Have you ever used the **Generic Text Driver** for printing? If not, you are missing considerable savings in printing time, especially if you are using slower printers. Usually, when printing a draft copy of your text, you want the quickest possible printing. The Generic Text Driver is the answer. If you set up your printer to use the Generic Text Driver, you can select it for quick draft printing whenever needed.

Note that the font selected for printing by the Generic Text Driver may not be exactly the same as in your topic.

Inserting date/time in your text

You can insert the **date** or **date/time** in your text by selecting the menu items **Edit-Insert date..**. You also have an option of selecting whether a long or short format date is inserted. WhizNotes picks up the format of the date and time from your current selections in the **International** settings of the Control Panel.

Toolbar/Statusbar

Toolbar:

Frequently used menu operations are available on a toolbar below the menu bar. To know the operation associated with a button, just move the mouse cursor over it and a hint on the operation will be shown on the status bar at the bottom.

Also, when you press a toolbar button, its menu equivalent will be shown on the status bar. *If you don't want to execute the operation, move the cursor out of the button before releasing it.*

Statusbar:

For many operations, a status message appears on the status bar. The message usually provides help on canceling a long running operation.

You can choose to view or not to view the toolbar and statusbar by the **options menu**.

Options menu

The options menu allows you to control the following entities:

Toolbar

Uncheck this menu item to remove the [toolbar](#). You may rarely want to do that since the toolbar is so useful.

Statusbar

Uncheck this menu item to remove the [statusbar](#).

General desktop options

See [General desktop options](#).

Printing options:

See [Setting up print margins and other printing options](#).

Spelling-Checker options:

Spelling-Checker is an add-on program that works only with the registered version.

All other menu items are meaningful only when you are working on a WhizFolder file or a topic. For details of those items, please see [Desktop options for a WhizFolder](#), or select the item on the menu and press F1 for getting help on it.

Setting up the Help Compilation environment

By default, WhizNotes sets up the environment to invoke the help compiler **HC31.EXE**. If you want to invoke a next version of the help compiler, for which the file name is different, please look at the following information. Otherwise, you may skip this topic.

Changing the name of the help compiler file that is used:

Let us assume that the name of the new released help compiler EXE file is **HCP.EXE**. To set up this compiler for use with WhizNotes, you must use Windows **File Manager** to **associate** the file extension **HPJ** with **HCP.EXE**. If the help compiler is not on your DOS path, it is better to type in the full path name of the help compiler in the File Manager association.

This works because WhizNotes has been programmed to use **HC31.EXE** only if you have not already set up the association of extension **HPJ** yourself. This means that WhizNotes will be able to invoke future versions of the help compiler, provided you make the proper association for the **HPJ** extension in the File Manager.



In Windows 95, you need to set up the association in Explorer by the menu **View-Options**. Select properties "file types" and see if "Help Project File" is in the list. If it's present, you can change its association. Otherwise, you'll have to add an item for the HPJ extension and specify the name of the help compiler EXE file.

Once you have set up this environment, you are ready to generate Windows help from your WhizFolder files.

Windows 95 based HCW compiler:

If you are using the new Windows 95 based HCW compiler, you need not worry. Its installation takes care to associate it with HPJ file, unless you changed the association. In that case, just run HCW and it will again install itself. This poses a problem if you want to switch between HCW and the old DOS based compilers. For a work around, please see [Using the new HCW compiler](#).

"Out of memory" error from the compiler

If you use HC31.EXE then you might experience memory shortage when compiling the help files containing large bitmaps. The solution is to get a protected mode version of the help compiler, HCP.EXE, that uses extended memory. See [Where to get the help compiler](#) for more details.

Also see:

[Where to get the help compiler](#)
[How to help-compile in background](#)
[Using the new HCW compiler](#)

Temporary files created by WhizNotes

WhizNotes needs to create and use temporary files when WZN files are open. You should know that the temporary files are created in the Windows temporary directory under a subdirectory named **WHIZTEMP**. Further, more subdirectories with names like **NTWORKAA.AAA**, **NTWORKAA.AAB**,... are also created in this **WHIZTEMP** subdirectory. You should never delete such directories when WhizNotes is running, as you may lose any changes that you are making.

Also see:

[Please check the program's speed](#)

Exporting selected topics to a text file

At a certain stage, after preparing your notes in WhizNotes, you may want to join one or more topics to create a single external text file. This is called an **export-text** operation.

To perform an export-text, please select the menu **Tools-Export selected topics to text file....** A dialog box appears that allows you to specify the following options:

Enter delimiter line:

You can enter a delimiter line that you want to precede the start of each topic in the output text file. For example, if you enter **/**TOPIC**/** as the delimiter line then WhizNotes puts such a line in the output text file whenever it starts writing a new topic. Of course, if you want, you may not choose to put any delimiter by leaving the line empty in the dialog box.

Delimiter is especially useful, if later you want to import the same file back into a WhizFolder. *Please also note that this is the only option which you should use on this dialog box if you foresee reimporting this text back into a WhizFolder file. If you use any of the following options then topics may not be imported exactly as they looked before exporting.*

Break paragraphs into lines at a specified column:

You can word-break paragraphs into lines at a specified column. This is very useful for preparing properly formatted files for MS-DOS viewing. For example, I often use this feature to prepare a README.1ST file from my topics.

Do not export topic name:

You may choose NOT to export topic names before the text. You may want to do this if you simply want to merge text from selected topics into one big text file. For example, you may have earlier imported a whole, big text file that was split into multiple topics, and now you want to export the pieces to recreate the whole file. See Importing whole text files as topics for how to import big text files into a WhizFolder.

Emit an underlining line of dashes:

Emit a blank line before and after the topic name:

These are additional features to nicely format the exported file for viewing with external commands like DOS **Type**.

Also see:

Importing topics from a text file

Importing topics from a text file

The **Import topics** dialog appears when you select the menu **Tools-Import topics from a file**. Import itself is a fairly advanced operation. You should clearly understand what happens here. Hence, the following discussion attempts to give you lots of details on finer points of importing. For a first reading, you may skip the details as necessary.

A delimiter line:

When importing topics from a text file, you must enter a delimiter line so that WhizNotes can separate out the topics. Once you specify a proper delimiter line, the import feature will import topics into the WhizFolder window.

Please note that if a WhizFolder has already been exported to a text file with delimiters then it is ready for import automatically. Try it out. See the topic Exporting selected topics to a text file for details on how to export.

If you are taking an external text file and want to break it into topics then, and only then, you need to do some extra work to insert your delimiter and topic names at the proper places in the file. Please also see Knowing a little programming helps when importing.

Join lines to form paragraphs:

This is an extremely useful feature when importing text prepared by non-word-wrap editors which introduce a hard carriage-return character at the end of each line of a paragraph. If you were to rejoin these lines manually in the topic window, it's a tedious process. WhizNotes uses a special technique to recognize the lines belonging to a paragraph. Of course, it may not always work, especially, if left margins are present in the text file in the form of spaces. That is why, after performing any import, you should examine the imported topics to correct any undesired joining of lines.

Those of you who are used to preparing a lists of items with a hyphen in the first column will be delighted to know that WhizNotes does not join items of such a list.

If the above join line operation does not work the way you want it to work, you should try the Convert into paragraph feature available on the edit menu of an open topic window. It is much more convenient and sophisticated.

Append topics with duplicate names:

What happens when a topic name to be imported already exists in the WhizFolder file? Well, you have two choices: the default behavior is that WhizNotes makes up a new topic name by appending a sequence number for the new topic. However, if you check this box, WhizNotes tries to append the new topic's text to the already existing topic. If it runs out of space for a topic, it creates another similar topic name and continues. This is very useful for making topics out of email messages based on the **subject** where a subject may repeat quite often.

Also see:

Importing whole text files as topics
Exporting selected topics to a text file

Importing whole text files as topics

This operation allows you to import external text files, each as a separate topic. If a file is larger than a predetermined topic size, it's split into multiple topics.

There are two ways to perform this import operation:

1. Select the menu item **Tools-Importing whole text files as topics...** A standard file open dialog comes up where you can select multiple files to be imported.
2. **Drag** multiple files from the **File Manager** or **Explorer** and **drop** them on the WhizFolder window.

You will see that files will be imported and split into multiple topics if they are large. You can use this feature for several purposes. I use it often to take advantage of the convenient WhizNotes printing. I can drop all my program source files on a WhizFolder window, rename them to more descriptive names and rearrange their order. Then I get a neat printout from WhizNotes with a table of contents at the end. You may find other uses for it as well.

Just to let you know, the large files are purposely split into multiple topics at less than maximum topic size, so that you can add more information to these topic-pieces. The sequencing of the pieces uses a special numbering convention with enough gaps, so that you can insert more pieces. Although you may not want to write these pieces back to form the original file, you can do so if you select them all and then export with a **blank delimiter line** and the option **Do not export topic name**. This is a WhizNotes work around that will enable you to edit very very big text files.

Also see:

[Importing topics from a text file](#)

[Exporting selected topics to a text file](#)

Exporting selected topics in RTF format

This is an excellent way to convert your help into a printable document.

Note: The term **non-help** means that WhizNotes does not include WinHelp hypertext links and footnotes in the document. Hence, the document becomes a clean printable document. *Hence, don't try to compile this RTF file for help.*

If you select one or more topics and select the menu **Tools-Export selected topics as non-help RTF** then WhizNotes writes the selected topics to an RTF file that has the same name as the WhizFolder file, but its extension is **RTF**. You can open this file in your favorite word processor and can further edit it as needed and take a rich format printout.

After the RTF file is created, WhizNotes also gives you an option to copy the RTF text to the clipboard so that you can directly paste it in your word processor.

Limits set in WhizNotes

At present WhizNotes has the following limits:

- Please note that WhizNotes is a single user program. Two users can not use the same WZN file on a network at the same time. WhizNotes safeguards this by opening WZN files in exclusive access mode so that two users are prevented from accessing the same file simultaneously. However, this won't be effective in some cases unless you install SHARE.EXE.
- A topic can contain up to about 32K characters of text. However, since WhizNotes encourages you to divide your document into topics of manageable size, this is not really a limit. In fact, you can have very large WhizFolder files made up of these topics.
- The topic name can be up to 255 characters in length. However, there is no restriction on using special characters, even blanks, embedded in topic names. If you are generating help, it is better not to include a semicolon in the topic name. If you do, the only problem is that the help keyword for the topic name will not be generated correctly.
- WhizNotes allows you to keep maximum 50 windows open at the same time but that can be changed by an INI flag. Please see [Less often used INI flags](#).
- The **list of topics** in a WhizFolder window can have maximum 32K topics.
- When editing external text files, the same limit of 32K characters applies.

A printout which is solely produced for checking the text content of a document. It is faster because it is usually produced by sending a stream of characters to your printer. There is no font selection involved.

Examples of WhizFolder files

A WhizFolder file is a collection of many topics. You create such files with the help of WhizNotes, and they have the extension ".WZN".

It's up to you to group any topics you want into one WZN or WhizFolder file. This way you don't have to remember each of your topic names. All you need to remember is the WhizFolder file name that serves as a folder name.

A topic within a WhizFolder file can be any independent text note that you want to recognize by name. Please also note that topic names appear in sorted order in the list of topics so that you can easily locate them by name. Of course, you have the option of rearranging the sequence of your topics if you uncheck the menu item **Edit-Sorted list of topics**.

Example:

A WhizFolder file "WEEKREPS.WZN" may contain various weekly reports as individual topics, having the following names:

- 01/02/93 - Weekly report (sent on 01/05)
- 01/09/93 - Weekly report
- ...

As you can see, the ability to choose any topic name and to group these topics makes WhizNotes extremely useful. You need not care about how the topics are kept or which directory they belong to. All you need to remember is the name of that particular WhizFolder file that contains them: "WEEKREPS.WZN" in this case.

More examples:

- A WhizFolder file "LETTERS.WZN" keeps your letters as individual topics, with names like "*Martin - letter sent on Dec 14*".
- A WhizFolder file "UTILSLIB.WZN" is used by a programmer to keep notes on various C utility functions or C++ classes that he is developing or plans to develop. Again the flexible naming scheme allows topic names like:

```
int  getDelimitedText(char *textBuffer)
```

- A writer uses a WhizFolder file "PHRASES.WZN" to quickly note any phrases or ideas he might find useful in a writing project; he may use topic names like "*Possible beginning para*".
- Two consultants, on a project, share ideas by passing a WhizFolder file called "prodidea.wzn" and thus exchange information on related topics.
- ...

Although, the above examples use short file names for the WZN files, note that you can use long file names on Windows 95. WhizNotes automatically detects the long file name system and allows you to store WZN files by long file names.

Isn't it wonderful to be able to identify all your information by such convenient topic names and to group them together in WhizFolder files?

It may be interesting for you to know that--as the WhizNotes author--I myself have been keeping track of enhancement ideas and bug reports using WhizFolder files. This way I never lose any single piece of information that I want to remember. WhizNotes allows you to convert any *WhizFolder file* into Windows hypertext help, provided you have the Windows help compiler. To find out more, please see [Basics of Help generation](#).

Features for organizing hypertext

WhizNotes helps you take maximum advantage of the random creative thought processes that occur when you are preparing a document, doing research or studying a new subject.

Let us look at the basic features for organizing hypertext:

Choose any name for your topic:

Flexibility in choosing topic names is one of the most convenient features of WhizNotes. After using this facility of descriptive and clear names, you will never go back to keeping your text in cryptic filenames on the disk.

With WhizNotes, you can make topics having very long names, up to 255 characters long. Also, there is no restriction on the set of characters that you can use in the topic names.

This flexibility in choosing names can be put to hundreds of different uses. For example, if you are a programmer you may as well make function prototypes as your topic names to facilitate direct access to the documentation for your function library.

Easily group related topics together:

WhizNotes also provides a convenient grouping facility. You can combine related topics into a single file. Such files are called WhizFolder files. One WhizFolder file may contain many topics and you can have different WhizFolder files for different subjects.

Easily locate a topic:

The WhizFolder file window lists topics in a sorted order. This makes it easier to locate a topic. You can choose to view the list of topics sorted by topic names or you can rearrange the topics in any desired sequence.

Search functions are also available to quickly locate topics or to search for text in topics.

Insert hypertext jumps between topics:

You can insert JUMP statements by dragging and dropping a topic name onto an open topic window. Then you can double-click on such jump statements to move to the inserted topic. Later if you convert your notes into help, these jump statements are converted into Windows hypertext jumps.

Drag and drop topics from one WhizFolder to another:

You can have multiple WhizFolder files open at the same time and you can copy topics from one WhizFolder file to another by just dragging and dropping the topic names.

Use flexible Multiple Document Interface:

WhizNotes provides the standard Multiple Document Interface similar to many applications. Hence, you can learn it quickly and start using it immediately.

Insert Jump Dialog Box

When you **drag** a topic from the list of topics onto one of the open topic windows then this dialog box appears. You can select a **jump** or **reference (popup)** to be inserted at the caret position. The dialog box remembers your preference so that next time you can just hit **Enter** to insert the proper statement.

If you prefer clipboard operations and want to *copy and paste* a jump statement, that method is available too from an open topic window (menu **Edit-Copy special...**) or from the topic list on a right-click.

What is a Jump statement?

Using this jump, you can literally jump to the indicated topic by double-clicking on such a statement. This is very convenient for navigating between topics.

A WhizNotes jump statement looks like this: [**~JUMP topicname**].

Also, when you generate a Windows help file from WhizNotes, this jump becomes a Windows help jump link.

What is a Reference?

The double-click on a reference also works similar to a jump but, in the generated help, this is converted to a popup link.

A WhizNotes reference statement looks like this: [**~REF topicname**].

Windows 95 tip

Use this tip only if you are using Windows 95 or a later Windows version.

Multiple Document Interface

WhizNotes uses the standard Multiple Document Interface **MDI** similar to many Windows applications. This similarity helps you learn it quickly and easily.

For instance, if you have used **Program Manager**, **File Manager** or **MS Word** then you are already somewhat familiar with **MDI**.

WhizNotes **MDI** allows you to:

- Keep multiple topics as well as WhizFolder windows open at the same time.
- Use a menu located consistently at the top of the main window regardless of the current window being used (what menu options you see are all relevant and applicable to the window that is in focus or active).
- Easily locate a window by the **Window menu**.
- Minimize or maximize windows as a convenience in organizing your desktop space.
- Tile or cascade windows.
- Arrange icons for minimized windows.
- *In addition to the standard MDI features, WhizNotes implements some enhancements. For example, you can close all the topic windows without closing the WhizFolder windows. You can also minimize all the topics to unclutter your desktop*

Printing all the topics of a WhizFolder file

If you select the menu **File-Print All Topics...**, the following actions take place:

You are asked for a title for your printout. All the topics in your current WhizFolder file are printed, and proper headers and footers are inserted in the printout by WhizNotes. In addition, a table of contents is printed at the end.

[Print list of topics](#) is also available on the File menu.

Also see:

[How jumps are printed](#)

[Printing from a topic window](#)

[Printing one or more selected topics](#)

[Setting up print margins and other printing options](#)

Select Topic Dialog Box

In this dialog box, as you type the beginning characters of a topic name, it selects the corresponding topic in the WhizFolder window that matches the beginning characters. When you press OK on this dialog, the selected topic is immediately opened for editing.

This feature works better when you are viewing the list of topics in a sorted order.

The subject matter or thoughts expressed in a document.

A WhizNotes jump is a hypertext jump statement that allows you to double-click on the jump and go to the topic window indicated.

What is a delimiter

When you want to import topics from a text file in such a way that certain portions of the file become separate topics then how can WhizNotes identify the boundaries of these topics? First of all, the text file should have multiple topics in such a manner that each topic name appears on a separate line by itself and the text for the topic appears on the following lines. But how can WhizNotes recognize the end of the topic text and the line containing the next topic name? For this purpose, the **delimiter line** is used. In other words, the text file must have an identifier-line preceding each topic name. Otherwise, it's not possible for WhizNotes to make out the end of one topic and the beginning of the next one. A delimiter line serves the purpose of the identifier-line that breaks up the file into topics.

Example:

Your text file may look like this:

```
/**A TOPIC**/  
First Topic Name  
...  
...text for the topic...  
  
/**A TOPIC**/  
Second Topic Name  
...  
...text for the topic...
```

Here, the line **/**A TOPIC**/** identifies that a new topic is starting, and the next line contains a topic name. Hence, the line containing **/**A TOPIC**/** is a delimiter line. A delimiter must be present in the text file to use the import topics facility.

Also see:

[Importing topics from a text file](#)

Knowing a little programming helps when importing

When you want to split external text files into the topics for importing into WhizNotes, it'll help a lot if you know a little programming, especially for large and routine files like email messages.

For example, suppose you have a collection of email messages where you want to make the subject a topic name. But the subject actually appears on the third line, whereas you want the topic itself to include the first line which contains FROM part of your email message. If you know some programming, you can write a program that takes the third line and prefixes it with a delimiter line before the mail message. Once you do that, your email messages can be imported as separate topics.

Even if you don't know programming, you can perform some tricks to insert delimiters and topic names with Windows accessory **Recorder** and a text editor that can edit large text files. The hint is to use the clipboard and also the keyboard commands for the editor so that you can perform operations like find some text, move back a line, skip the first word, select rest of the words, copy them to clipboard, etc.

Also see:

[What is a delimiter](#)

[Importing topics from a text file](#)

How to insert bitmaps into your help files

You can insert bitmaps (pictures) into your help text with the following bitmap statements:

[~ **Bitmap bitmapfilename**]

Inserts the bitmap at the current position in the text.

[~ **BitmapLeft bitmapfilename**]

Left justifies the bitmap. Any following text wraps around this bitmap on the right side.

[~ **BitmapRight bitmapfilename**]

Right justifies the bitmap. Any following text wraps around this bitmap on the left side.

You can also use **.shg** (hypergraphics) and **.mrb** (multi-resolution bitmaps) in the above statements. If you use large bitmaps and face memory shortage, you should consider updating to a protected mode compiler or to the new HCW compiler (for Windows 95). For details, see [Where to get the help compiler](#).



If you're using the new HCW compiler, even the **.dib** and **.wmf** files are also allowed in the bitmap statements.

Avoiding bitmap distortions:

When using a different screen resolution, the bitmaps in your help file may seem distorted. Why?

Usually, WinHelp tries to maintain a bitmap's logical size across displays by stretching or shrinking the bitmap when displaying it. This might distort certain bitmaps. There's another problem. If you include a 256 color graphic, it might look bad on a 16 color monitor.

If you want more control on how the bitmaps are displayed on different devices, you need to do some more work:

Solving the screen resolution problem:

You need to create bitmaps for different screen resolutions and combine them into a single graphic file. An MRBC (Multi-resolution bitmap compiler) tool in your SDK can do that. It creates a .MRB file. Then, all you need to do is, reference this .MRB file in your bitmap statements. When WinHelp encounters a Multi-resolution bitmap, it selects the bitmap that most closely matches your display device.

Solving the color problem:

There is no adequate solution for color problems in Windows 3.1 help files. You should rather use optimized 16-color images which will work well over a wide range of platforms.



In the help files (V4.0) created with HCW, you can be smarter. You can simply create different color images in separate files and specify all of them in the above bitmap statement, separating them with semicolons. WinHelp 4.0 will use the one which matches the color capability of the monitor. However, if you use the same bitmap again, be sure to specify the images in the same order, otherwise it will take up more space in your help file. You can use RtfMacros feature to avoid this problem. Just specify the bitmap statement in an RtfMacro representing that bitmap and use it everywhere.

Bitmaps as hotspots:

In Windows help, you can have hypertext links from a bitmap. These are called hotspots. When the mouse cursor is over the hotspot, it becomes a hand symbol showing a hypertext link. Using bitmap hotspots can make your help file much more attractive.

If your bitmap contains only one hotspot, it's better to use [Bitmaps as hotspots](#).

For multiple hotspots, we call it hypergraphics. See [Bitmaps as hypergraphics](#).

Making bulleted and numeric lists

Bulleted lists:

To make up the bulleted lists in your help text, you can use the following statements, **each on a separate line** by itself:

- Use [**~StartList**] to start the list.
Use it on a separate line by itself.
- Use [**~Item**] on the preceding line before each item.
Use it on a separate line by itself. Item can be as long as you want, even consisting of multiple paragraphs. The indentation will be retained for the whole item.
- Use [**~EndList**] to end the list.
Use it on a separate line by itself.

For example, the above bulleted list was created by using the statements as described. You can even nest lists within lists!

Numeric lists:

To specify a numeric list, simply use [**~StartList Numeric**] instead of just [**~StartList**].

Note that the above numeric list uses numbers with dots as suffixes. There's another variation available in the Windows 95 help style which removes the dots and makes the list compact. To do that, use [**~StartList NumericCompact**].

Changing the list indent size:

WhizNotes uses a list indent value as half of the tabsize specified on the help generation dialog (0.5" or same as WhizFolder, depending on your selection).

Indenting text

Sometimes, you would like to indent blocks of text in the generated help. To do that, use the following type of statement on a separate line by itself:

- Use [**~Indent+**] to indent one step to the right.

Any paragraphs after this statement will remain indented. You can repeat this statement to move the indentation to the right, one step at a time.

- Use [**~Indent-**] to move back the indentation one step to the left.
- Use [**~Indent0**] to remove all indentation.

Use this statement when you want to make the text appear as it was originally, without indentation. Note that a zero has been used instead of + or -.

Changing the indent size:

WhizNotes uses an indent or list-indent value as half of the tabsize specified on the help generation dialog (0.5" or same as WhizFolder, depending on your selection).

Global options to control help generation in general

To control the help generation, you may require use of certain global help generation options provided by WhizNotes.

What you have to do is, make a special topic named **~HelpAttributes** and define a **GlobalHelpOptions** statement using the statement syntax given below. Only one **GlobalHelpOptions** statement is allowed.

Format of GlobalHelpOptions:

```
GlobalHelpOptions
{
    ContentsTopicName = "Contents",
    ForceHelpCompilerToUse = "",
    RTFBeforeTopicTitles = "",
    RTFAfterTopicTitles = "",
    StartingContextNumber = ""
    CustomHpiFile = "",
    CustomRtfHeaderInfo = ""
}
```

Tip: A note on quoted values

Here's an explanation for the various options:

ContentsTopicName:

ContentsTopicName allows a foreign language user to specify a different topic as the Contents topic name. The default contents topic generated by WhizNotes will have this name. Also, if such a topic is present, WhizNotes will interpret the contents of that topic as the help **Contents**.

ForceHelpCompilerToUse:

Normally, WhizNotes uses the help compiler installed on your system which is the one associated with HPJ extension in your File Manager or system registry. But you can force WhizNotes to use another compiler by naming it in this option. For example, if you're using Windows 95 and new HCW is installed on your system. If you want to generate help using the old HC31.EXE, then you have to put ForceHelpCompilerToUse = "HC31.EXE". You may ask, what if I install HC31.EXE by associating it with HPJ extension? The answer is, it will work but will stop working as soon as you start new HCW even once. The HCW has a habit of associating itself to the HPJ extension whenever it's started.

Note that if you use the "Just compile..." option on the help generation dialog then this compiler can't be forced. In that case, the compiler associated with HPJ extension is used.

RTFBeforeTopicTitles and RTFAfterTopicTitles:

If you want to customize how the topic titles are formatted, you can code these options. For example, you may want to change the color, make the font bigger, or insert a bitmap before the titles. You need to have a knowledge of RTF and you also need to replace the RTF file header by your own.

StartingContextNumber:

You can use this attribute to control the range of context numbers generated for the help topics in C header file (**INC file**). You may want to do this when you are providing context sensitive help and want to force assignment of context numbers to some help topics by using a HelpTopic statement. If you do not use this attribute then the default value used by WhizNotes is the constant **256** (which is same as **hexadecimal 100** used in earlier versions of WhizNotes).

CustomHpjFile:

If you specify an HPJ file name using this option then WhizNotes doesn't generate an HPJ file but rather copies the HPJ file given by you to create an HPJ file. This can be useful to add statements to HPJ file for interfacing with third party DLLs and RTF files generated by other systems. However, now more powerful customization options are available so that you can code sections in WZN file to be generated by WhizNotes. For details, see Customizing Help Project (HPJ) file.

CustomRtfHeaderInfo:

If you are an RTF expert, this option allows you to customize the RTF header so that you could use other fonts and colors in the help text. You may find an example of customizing the RTF header in an additional sample file.

Also see:

A note on quoted values

Window menu

Window menu allows you to organize your WhizNotes desktop by providing the usual tools, such as, 'Tile', 'Cascade', 'Arrange icons', 'Close all windows', 'Next window' and 'Previous window'. You also get some special menu items. These are 'Tile horizontal', 'Close all topics' and 'Minimize all topics'.

Also, a list of open windows is attached to the Window menu so that you can locate an open window easily.

General desktop options

This dialog comes up when you select the menu item **Options-General desktop options....** It allows you to specify certain desktop options which are general and do not belong in the options for individual WhizFolders.

Here's a description of the items available on this dialog.

Options for text file editor:

These options allow you to specify the **tabsize** and **font** for the editor windows used to view external text files. Since the text files themselves have no space to remember the font and tabsize, this is the place where you can specify these attributes and WhizNotes will remember them.

Restore desktop on startup: (only available in the registered version)

If you check this, the complete desktop is restored when you start WhizNotes next time.

No confirm for reorder-topics:

If you check this, no confirmation to move appears when you rearrange the topic list with a drag and drop.

No confirm for move to another WZN:

If you check this, no confirmation to move appears when you move topics from one WhizFolder to another with a drag and drop.

Obtaining word count

You can obtain the word count on the selected topics by the menu **Edit-Give word count**. The word count is also available from an editor window where it applies to all the text present in that window.

The word count is shown in a message box. You can copy the contents of the message box to the clipboard by selecting the message text and pressing standard key combinations **Ctrl-C** or **Ctrl-Insert**. You can then paste this text in a file to keep track of how your word count progresses and how much writing you are able to accomplish.

What is word-wrapping?

You should always keep the word-wrap **ON** for all your documenting needs, including the help authoring. If you turn word-wrap **ON**, you don't have to press **Enter** key to go to the next line. The text automatically wraps around the right edge of the window if the last word can not fit. This is convenient so that you can enter a whole paragraph without pressing **Enter**. Please remember that if word-wrap is **ON**, the only time you hit **Enter** key is to start a new paragraph or to insert blank lines.

You may need to put word-wrap **OFF** only when editing the program source files and other external text files which will look ugly with the word-wrapped text. For example, if you are preparing a **readme.txt** file, to be viewed by DOS command **Type**, you should turn word-wrap **OFF**.

New WhizFolder

Creates a new WhizFolder window. A WhizFolder window is a folder where you can keep a collection of your notes or topics. This operation is also available on the toolbar.

Also see:

[What is a topic?](#)

[What is a WhizFolder file?](#)

[Creating a new WhizFolder file](#)

Open WhizFolder...

Presents a file-open dialog to let you open a WZN or WhizFolder file that you created earlier. This operation is also available on the toolbar.

Also see:

[What is a topic?](#)

[What is a WhizFolder file?](#)

[Opening an existing WhizFolder file](#)

[Opening from File Manager](#)

Save WhizFolder

Saves the current WhizFolder file along with all of its topics. Even the changed topics are force-saved without asking you for a confirmation to save. *Note that a Save is not performed if nothing has changed.* **Hence, when working on a WhizFolder window, you should perform this operation from time to time to save your changes.** This operation is also available on the toolbar.

Also see:

[Saving a WhizFolder file](#)

Save WhizFolder as...

Allows you to save the current WhizFolder file by another name.

Also see:

[Saving a WhizFolder file](#)

Close window

Closes the currently active window.

Close WhizFolder

Closes the currently active WhizFolder window along with all the child topics. This operation is also available on the toolbar.

Close parent WhizFolder

Closes the the parent WhizFolder of this topic along with all the child topics. This operation is also available on the toolbar.

Save all

A save is performed for all the WhizFolder, topic and text file windows which have changed since the last save. This operation is also available on the toolbar.

Open selected topics

Opens one or more selected topics for editing.

Also see:

[What is a topic?](#)

[Opening a topic for editing](#)

Rename topic...

Allows you to rename the current topic.

Print selected topics...

Prints selected topics after asking you for some preferences.

Also see:

[How jumps are printed](#)

[Printing one or more selected topics](#)

[Operations on multiple topics](#)

[Print list of topics](#)

Print all topics...

Prints all the topics after asking you for some preferences. This operation is also available on the toolbar.

Also see:

[How jumps are printed](#)

[Printing one or more selected topics](#)

[Print list of topics](#)

Print...

Prints the text from the current window. This operation is also available on the toolbar.

Also see:

[How jumps are printed](#)

[Printing from a topic window](#)

Open text file...

Presents a file-open dialog to let you open an external text file for viewing or editing. This operation is also available on the toolbar.

Also see:

[Manipulating external text files](#)

New text file

Allows you to create a new text file. This operation is also available on the toolbar.

Also see:

[Manipulating external text files](#)

Force word-wrap for selected topics...

Allows you to set word-wrap **ON** or **OFF** for selected topics.

For instance, after importing a number of topics from a text file you may want to change word-wrap attribute for all of them at once. If this option were not there you would have to individually open each topic and change its word-wrap attribute.

Also see:

[Operations on multiple topics](#)

Word wrap

Allows you to turn word-wrap **ON** or **OFF** for this topic. Check this menu item if you want the word-wrap **ON** and uncheck it if you want the word-wrap **OFF**.

Also see:

[What is word-wrapping?](#)

Select all

Selects all the text in the current window. If the current window is a WhizFolder window, all the topics in the list are selected.

Insert date/time

Inserts current date and time at the caret position. You can control the format of the date and time by properly setting up your choice in the **International** settings under Control Panel.

Insert date

Inserts current date at the caret position. You can control the format of the date by properly setting up your choice in the **International** settings under Control Panel.

Copy selected names

Copies the names of selected topics to the clipboard so that you can paste them elsewhere.

This action is also available on the Right-click menu on the topic list and from an open topic window's Edit menu.

Also see:

[Getting WhizFolder details by right-click](#)

Sorted list of topics

Uncheck this menu item if you want to change the order or sequence of the topics in the list. Check it to view the list in alphabetically sorted order.

Note that the **move topic** operations are only allowed on an unsorted list of topics.

Also see:

[Ordering your topics in the topic list](#)

Move selected topics up

Moves the selected topics up in the list. If you are using a mouse, it is more convenient to use the **arrow buttons** on the right of the topic list. These arrow buttons represent move to Top, Up, Down and Bottom respectively.

The moved topics appear bunched together after the move. A move is allowed only when you are viewing the list in unsorted order.

Using a mouse, you can even drag and drop topics to reorder them.

Also see:

[Ordering your topics in the topic list](#)

Move selected topics down

Moves the selected topics down in the list. If you are using a mouse, it is more convenient to use the **arrow buttons** on the right of the topic list. These arrow buttons represent move to Top, Up, Down and Bottom respectively.

The moved topics appear bunched together after the move. A move is allowed only when you are viewing the list in unsorted order.

Using a mouse, you can even drag and drop topics to reorder them.

Also see:

[Ordering your topics in the topic list](#)

Move selected topics to top

Moves the selected topics to the top of the list. If you are using a mouse, it is more convenient to use the **arrow buttons** on the right of the topic list. These arrow buttons represent move to Top, Up, Down and Bottom respectively.

The moved topics appear bunched together after the move. A move is allowed only when you are viewing the list in unsorted order.

Using a mouse, you can even drag and drop topics to reorder them.

Also see:

[Ordering your topics in the topic list](#)

Move selected topics to bottom

Moves the selected topics to the bottom of the list. If you are using a mouse, it is more convenient to use the **arrow buttons** on the right of the topic list. These arrow buttons represent move to Top, Up, Down and Bottom respectively.

The moved topics appear bunched together after the move. A move is allowed only when you are viewing the list in unsorted order.

Using a mouse, you can even drag and drop topics to reorder them.

Also see:

[Ordering your topics in the topic list](#)

Use long date format

Allows you to use the long date format when inserting the date using the menu items **Edit-Insert date** and **Edit-Insert date/time**.

The long date and short date formats are selectable in the **International** settings of the Windows Control Panel.

Give word count for selected topics

Displays the total number of words in the text of selected topics. The count also includes the topic names.

Also see:

[Operations on multiple topics](#)

Give word count all

Displays the total number of words contained in all the text of this window.

Select topic beginning with...

Allows you to position to a topic quickly by typing the beginning characters of the topic name.

Also see:

[Searching for a topic](#)

Test Help

Invokes the help file that you last generated from this WhizFolder.

Export selected topics to text file...

Allows you to join a number of topics to form a single text file. This operation is also available on the toolbar.

Also see:

[Exporting selected topics to a text file](#)

Export topic to text file...

Allows you to export the current topic to a text file. This operation is also available on the toolbar.

Also see:

[Exporting selected topics to a text file](#)

Import topics from text file...

Allows you to import the topics from a specially formatted text file. This operation is also available on the toolbar.

Also see:

[Importing topics from a text file](#)

Import whole text files as topics...

Allows you to import external text files, each as a separate topic. This operation is also available on the toolbar.

If the files are larger than a pre-determined topic size, they are split into multiple topics.

Also see:

[Importing whole text files as topics](#)

Export selected topics as non-help RTF...

This is an excellent way to convert your help into a printable document.

Note: The term **non-help** means that WhizNotes does not include WinHelp hypertext links and footnotes in the document. Hence, the document becomes a clean printable document. *Hence, don't try to compile this RTF file for help.*

Also see:

[Exporting selected topics in RTF format](#)
[HHML](#)

General desktop options...

Allows you to specify certain desktop options which are general and do not belong in the options for individual WhizFolders. This operation is also available on the toolbar.

Also see:

[General desktop options](#)

Desktop options for this WhizFolder...

Allows you to change the options for viewing, controlling and organizing the current WhizFolder. This operation is also available on the toolbar.

Also see:

[Desktop options for a WhizFolder](#)

Reset selected topics to default sizes

If you have customized positions for your topics by moving or resizing them, you can force them to become normal topics by using this menu item. This is only enabled if you have selected the option 'Open topics at their last customized position/size' in the Desktop options for this WhizFolder.

Also see:

[Desktop options for a WhizFolder](#)

Toolbar

Check this menu item to see the toolbar at the top of the main window. Uncheck it to remove the toolbar.

Also see:

[What is toolbar?](#)

Statusbar

Check this menu item to see the status bar at the bottom of the main window. Uncheck it to remove the status bar.

Also see:

[Toolbar/Statusbar](#)

Prepare distributable WZN with latest options

Forces the saving of WhizFolder options in the WhizFolder file so that you can distribute it to another user.

For details, see [Desktop options for a WhizFolder](#).

Close all topics

Closes all the open topic windows.

Minimize all topics

Minimizes all the open topic windows. This clears some space on your desktop and you can have a better view of other non-topic open windows: WhizFolders and text files.

Tile horizontal

If you have **only two** windows open then **Tile horizontal** is convenient to view them together, horizontally aligned.

Previous

Moves to the previous open window.

Next

Moves to the next open window.

Help Contents

Displays Contents of the online help. This operation is also available on the toolbar.

How to use Help

Displays help on **How to use the online help**.

Search Help for...

Allows you to enter a **keyword** and see the help related to that **keyword**.

For example, to see the help related to **Drag & drop**, select this menu item and type **Drag & drop**.

Introduction to WhizNotes

Shows you the [Introduction](#) help topic so that you can know the concepts of WhizNotes.

Quick Start Tutorial

The following steps will help you learn the main features of WhizNotes as a Hypertext Organizer and a Help Authoring System. After that, start using it and explore the Online Help as you go. *It would be convenient if you print the contents of this help topic before performing the steps so that you don't have to switch between WhizNotes and this help.*

Step 1. Creating a WhizFolder and adding topics to it:

- Select the menu **File-New WhizFolder** to open a new WhizFolder window. You will see that a gray window appears that contains an empty list of topics; such a window is called a WhizFolder window.
- To add topics to this empty WhizFolder window, select the menu **File-Add Topic** or simply press the Insert key. You will be asked for a topic name. Enter any topic name, for example, "This is my first topic".

You will see that an editor window appears that allows you to enter the text for the topic. Try entering some text. Now close this topic. Add a few more topics by following the same procedure. You will see that they are automatically sorted when they appear in the list.

- You can open one or more topics for modification. To do that, select the topic that you want to modify by clicking on it. Then either press the Enter key or double-click on the topic name. You will see that the topic-text that you earlier entered appears in an editor window where you can modify it.
- Try opening a number of topics and do not close their windows. Select the **Window** menu. You will see a list of all the open windows. You can recognize the currently active window by a check mark that appears next to its name. You can make any other window active by selecting it from the list or you can directly click on the desired window to make it active. Try using other features on this Window menu. For example, try tiling, cascading, closing and minimizing operations.
- Now select the menu **File-Save WhizFolder**. You will have to select a name for this WhizFolder file. Select name "TUTORIAL.WZN", then close the WhizFolder window. All your topics are now saved under the file TUTORIAL.WZN.
- To reopen the file TUTORIAL.WZN, select File menu. You will see that the name TUTORIAL.WZN appears at the bottom of the menu. This is for your convenience so that you can quickly open this last used file. Just click on it to open. See that the topics are listed when the window comes up.

Step 2. Copying topics from one WhizFolder to another:

Let us open a new WhizFolder and copy some topics to it.

Select the menu **File-New WhizFolder** to open a new WhizFolder window. Click on 'Tutorial.Wzn' window to make it active. Select multiple topics by clicking on topics with the **control** key pressed. Now drag the mouse cursor while keeping the mouse button pressed. You will see that the cursor changes to the shape of multiple notes. Move this cursor to the second empty WhizFolder window while keeping the mouse button pressed. Release the mouse button when you are on top of the empty window. This will drop the selected topics on the empty window. You just saw a demonstration of drag and drop copying. Now the use of this second window is over so discard it by closing it and saying "No" to the save prompt.

Step 3. Inserting hypertext jumps between topics:

Let us try inserting a jump from one topic to another.

Open one of your topics from TUTORIAL.WZN. Position caret (blinking cursor) at a place where you want to insert the jump. Now drag another topic as described earlier but this time drop it on the open topic window. You will see that a JUMP statement gets inserted in the topic. What is the use of this jump statement? Well, double-click anywhere on this JUMP statement and see what happens. You will instantly go to the topic window indicated in the JUMP. Isn't it convenient!

Step 4. Creating Windows help:

If you are interested in help generation, don't forget to see [Sample WZN file to demonstrate Help Generation](#) to get a quick demonstration.

Step 5. Exporting RTF format

Open the sample WhizFolder file SAMPLHLP.WZN, and select the following topics by keeping Ctrl key pressed and clicking on the topic name:

- Making your text bold, italic, etc.
- Example of indentations
- List statements

Select the menu **Tools-Export selected topics as non-help RTF**. WhizNotes will create an RTF file named SAMPLHLP.RTF and will ask you whether you want to copy the RTF text to the clipboard. Say yes. Now open a document in Microsoft word (or another word processor that supports RTF). Paste the contents of the clipboard into the document. You will see that properly formatted text will appear in the document. You can also open the file SAMPLHLP.RTF to get the same result. Look at how WhizNotes formatting statements were used in the topics that you selected. These statements were earlier only available to the help users. Now you can also use them. Look at other formatting statements under help related topics and you are all set to create well formatted documents.

Refer to online help frequently:

This is enough to get you started; but remember, we have barely scratched the surface of WhizNotes' abilities. Be sure to look at the contents in this help and explore other features as they interest you. You will discover more and more features that you can use to your advantage.

New features

Shows you a list of new features.

Copy topic contents

Copies the contents of the selected topics to the clipboard. This operation is also available on the toolbar.

Also see:

[Operations on multiple topics](#)

Copy

Copies selected text to the clipboard. This operation is also available on the toolbar.

Paste new topic contents...

Allows you to add a new topic containing the text from the clipboard. This operation is also available on the toolbar.

Paste

Pastes text from the clipboard at the current position. This operation is also available on the toolbar.

Clear all

Clears or deletes all the text in the window.

Undo

Allows you to undo the effect of the last edit operation.

Find text in topic names/contents...

Allows you to search for text in the topic names and also in the topic contents. Starts the search forward from the current selection. This operation is also available on the toolbar.

When the text is found in one of the topic's contents, the topic window is opened up showing the found text. You can continue the search by pressing **F3** or by selecting menu item **Edit-Find next**.

Also see:

[Search/replace operations](#)

Find...

Allows you to search for text in the window. Starts the search forward from the current position. This operation is also available on the toolbar.

Also see:

[Search/replace operations](#)

Replace text in topic names/contents...

Allows you to search and replace text in the topic names and also in topic contents. Starts the search forward from the current selection. This operation is also available on the toolbar.

When the text is found in a topic, the topic window is opened showing the found text. Then a replace operation is performed on the text.

Also see:

[Search/replace operations](#)

Replace...

Allows you to search and replace text in the window. Starts the search forward from the current position. This operation is also available on the toolbar.

Also see:

[Search/replace operations](#)

Find next occurrence of text in topic names/contents...

Continues the previously specified search/replace operation from the current position. This operation is also available on the toolbar.

Also see:

[Search/replace operations](#)

Find next

Continues the previously specified search/replace operation from the current position. This operation is also available on the toolbar.

Also see:

[Search/replace operations](#)

Save topic

Saves the changed text of this topic for use in the next operation. Save is done only if the topic contents have changed since the last save. This operation is also available on the toolbar.

Note that **Save topic** does not actually save the topic into the WhizFolder file. Rather, the save confirms the changes so that if you close WhizFolder, the changes for this topic will be taken without a confirmation from you.

If you really want to save all the changed topics to the WhizFolder file, you should select **Save WhizFolder**. If you want to save all the open topics of all the open WhizFolder files and all the open text files, then select the menu **Save all**.

Save file

Saves the file. Save is done only if the text has changed since the last save. This operation is also available on the toolbar.

Save to text file...

Saves the topic to an external text file.

Also see:

[Saving topic to an external text file](#)

Save as...

Allows you to save the file by a different name.

Tile

Tiles all the open windows so that you can view them side by side for convenience. Note that the minimized windows are ignored during this operation.

This operation is also useful when you want to drag and drop topic names on the open topic windows to insert jump statements.

Cascade

Cascades all the open windows so that you can view all the window titles and can conveniently locate any window. Note that the minimized windows are ignored during this operation.

Close all windows

Closes all the windows.

If the contents of any window have changed, you are asked for a confirmation to save the contents before closing.

Exit WhizNotes

An exit from the program WhizNotes takes place and the program terminates after closing all the windows. This operation is also available on the toolbar.

If the contents of any window have changed, you are asked for a confirmation to save the contents before closing.

Quickly opening a remembered file

Opens the file name selected on the menu.

WhizNotes remembers last 8 files that you have used at the bottom of the **File** menu. This is for your convenience so that you can quickly open a recently used file by selecting its name on the menu.

Locating a window by Window menu

Activates the window with the selected title.

WhizNotes presents the titles of open windows in a list at the bottom of the **Window** menu. This is for your convenience so that you can quickly locate a window by its title.

Windows help internally uses special jump-identifiers for the help topics. Such identifiers are called **context strings**.

For more information, please see [Context strings](#).

Context strings

In many other help authoring systems, you have to think of a **context string** to be assigned to your help topic and remember it for use in hypertext links from other topics. Context strings are internally assigned with [Rtf-footnotes](#).

WhizNotes internally manages the context string assignment and substitution for you so that you don't have to worry about this tedious process. Also, WhizNotes generates a context string based on the topic name itself so that context strings are easier to recognize for use by programmers who may want to use them in their programs to invoke the online help. To see the WhizNotes generated context strings, please select the option to **generate include file containing context-ids** when creating help. The resulting **.INC file** will list the context string identifiers for each topic.

You also have an option of overriding a context string for special purposes. For details, please see [Keywords, macros, and other characteristics of topics](#).

FYI, if two topic names are quite similar, their context string derived by WhizNotes may be identical. In such a case, it resolves the clash by suffixing a number to one of them. If you want to resolve these clashes yourself, you can do that too. See [Resolving context string clashes](#) for details.

- **Flags:** Quoted values with "YES" or "NO" are flags. Instead, you can also use "Y" or "N" for these values.
- **Double-quotes in quoted values:** To use them you must repeat them for each single occurrence.
- **Continuation rules for quoted values:** Please note that with the exception of ListBrowseGroup statements, all quoted values can be continued on the next line by ending the double-quotes on the preceding line and starting them again on the next line. This is for your convenience.

This continuation rule only applies to the attributes or options defined in special topics **~HelpAttributes** and **~RtfMacros**. It does not apply to the text in your regular topics where a quoted value must be contained in one paragraph only.

About WhizNotes...

Menu **Help-About WhizNotes...** shows the About box for WhizNotes program. An **about box** shows you the version number of this program and other useful information.

Add another topic...

Allows you to add another topic to the parent WhizFolder of the current topic, then presents a new topic window so that you can enter the topic text.

Also see:

[What is a topic?](#)

[Creating a new topic](#)

Add topic...

Allows you to add a topic to the current WhizFolder, then presents a new topic window so that you can enter the text for the topic. This operation is also available on the toolbar.

Also see:

[What is a topic?](#)

[Creating a new topic](#)

Arrange icons

Rearranges the icons of minimized windows.

If you have moved the icons to arbitrary positions and the desktop looks cluttered, you can use this feature to tidy up your work space

Auto indent

Allows you to turn auto-indent **ON** or **OFF** for this topic. Check this menu item if you want the auto-indent **ON** and uncheck it if you want the auto-indent **OFF**.

Also see:

[What is auto-indent?](#)

Basics of Help generation

If you have the Windows help compiler, you can convert your WhizFolder files into Windows hypertext help with just a click on the menu **Tools-Create Help....** Creating jumps from one topic to another is as easy as a drag and drop. If you have ever tried to create Windows Help, you already know how tedious the whole process is. WhizNotes makes this process incredibly simple.

Don't forget to look at the [Terminology of help generation](#) to know the help features available in Windows help and WhizNotes. Also see [Advantages of using WhizNotes for help generation](#).

To get a quick demonstration of Help Generation, see [Sample WZN file to demonstrate Help Generation](#).

You must have a Windows Help compiler:

To take advantage of WhizNotes help generation, you must have the Microsoft help compiler. If you already have a Windows SDK or an equivalent, chances are that you already have the help compiler. See your SDK documentation for more details. If your help compiler filename is different from **HC31.EXE** then please look at [Setting up the Help Compilation environment](#) to see how to set up the environment to invoke the correct help compiler.

If you do not have the help compiler, see [Where to get the help compiler](#).

Prepare your help text to take advantage of Windows help:

If you have inserted Jumps or Refs from one topic to another, WhizNotes converts them to proper Windows Help jumps and references (popups). This is very convenient. For details, see [Inserting a jump to another topic](#). There are many other [HHML](#) statements that allow you to format your text for proper viewing under help and also to define other type of hypertext links.

Generating Windows Help:

To create a Windows help file from your WhizFolder window, select the menu **Tools-Create Help**. A dialog box appears that allows you to select various options for generating a Windows help file. For details, see [Help Generation Dialog Box](#).

After you have specified your preferences on the above dialog box, the help compilation is started. When the compilation terminates, your help file would have been created with the same filename as your WhizFolder file but with an extension of **HLP**. WhizNotes gives you an opportunity to test this generated help. You can also test this help at any time by selecting menu **Tools-Test Help**. You may also directly test the help file by double-clicking on it in the File Manager or Explorer. Please note that once you have created a **HLP** file, it is ready to be distributed to your customers and can be used on any PC where Windows is installed. Note that help files compiled with the new HCW compiler can only be used on Windows 95 systems.

Also see:

[Terminology of help generation](#)
[Setting up the Help Compilation environment](#)
[Help generation with WhizNotes](#)

HHML

Basics of WhizNotes printing facilities

If you select a **File-Print...** type menu operation, WhizNotes selects a suitable printing format depending on which type of window is active.

Printing from a WhizFolder window:

When you are viewing a list of topics, you have the option of printing the selected topics or all the topics or a list of topics. WhizNotes introduces special headers and footers for identifying a particular topic in the printout. A table of contents is also printed at the end. You can control some aspects of this printout by the menu **Options-Printing options....**

Printing from an open topic window:

This is the feature that you will use most often when printing a single topic, say a letter. No special headers or footers will be printed except for the page numbers.

Forcing page-breaks:

Use a [**~PAGE**] statement in your text anywhere to force a page-break. Also, if you are viewing external text files and they contain Form-Feed control characters (^L) then WhizNotes will perform a page- break at such characters.

Also see:

[How jumps are printed](#)

Create Help for the owner WhizFolder...

Creates Windows hypertext help from your notes, provided you have the Microsoft help compiler. All the topics from this WhizFolder are included in the generated help. This operation is also available on the toolbar.

Also see:

[Basics of Help generation](#)

Create Help...

Creates Windows hypertext help from your notes, provided you have the Microsoft help compiler. This operation is also available on the toolbar.

Also see:

[Basics of Help generation](#)

Creating a new topic

Creating a new topic is simple:

- If a WhizFolder window is active, just press **Insert** or select the menu **File-New topic....**
- If a topic window is active, just press **Insert** or select the menu **File-Add another topic....**

A dialog box appears that allows you to choose a name for your topic. Please note that embedded blanks and special characters are allowed in topic names. So you can even choose a phrase for your topic name.

Once you select a topic name for your new topic, an editor window comes up that allows you to enter text into your topic.

You can also directly paste the clipboard contents as the contents of a new topic by selecting **Edit-Paste new topic contents**. You're asked for the topic name and the topic is created with the pasted contents.

Also see:

[Using 'Separator' topics to organize the topic list](#)

Creating a new WhizFolder file

To create a new WhizFolder file, select the menu **File-New WhizFolder**. You will see that an untitled gray window appears that contains an empty list of topics. This is called a WhizFolder file.

You can add one or more topics to this WhizFolder file.

Please note that WhizNotes allows you to keep multiple WhizFolder file windows open at the same time. You can even copy topics by dragging them from one WhizFolder file to another.

Also see:

[What is a WhizFolder file?](#)

[Creating a new topic](#)

[Saving a WhizFolder file](#)

[Save WhizFolder as.. menu option](#)

Cut

Cuts selected text to the clipboard. This operation is also available on the toolbar.

Delete

Deletes the selected text in the window.

Delete selected topics

Deletes selected topics after a confirmation from you.

Also see:

[Deleting a topic](#)

[Operations on multiple topics](#)

Delete topic

Deletes this active topic after a confirmation from you.

Also see:

[Deleting a topic](#)

Deleting a topic

To delete a topic, just select the topic and click on the menu **File-Delete Topic** or press the key **Del**.

You can select multiple topics in the list of topics and delete them. For details on how to select multiple topics, please see [Operations on multiple topics](#).

Delete operation is also available from an open topic window. But the keyboard shortcut is **Alt-Del** combination.

Desktop options for a WhizFolder

Here are the options that you can specify to change certain behavior of a WhizFolder window. This dialog comes up when you select the menu item **Options-Options for this WhizFolder...**

Title for WhizFolder window:

If you enter a title here, the filename in the title will be replaced by the title that you entered. This may be more convenient to identify the contents of your WhizFolder file. However, it's better to prefix the title with "WZN" so that you can recognize it in the list of Windows on the Window menu.



You may not need to use a title on Windows 95 as you can give long file names to identify your WhizFolder file. But it might still be useful if your complete long file name is too long.

Characteristics of topic windows:

These options control how the topic windows operate. For example, you can specify the tabsize and font for the topic windows. Note that the tabsize and font applies to all the topics of the current WhizFolder only.

Choices for viewing topics:

- **Open topics in normal cascaded manner:**

Normally, when you open a WhizFolder file and its topic windows, they are opened in a pre-determined cascaded position. This is convenient in most of the cases.

- **Open topics at their last customized position/size:**

Sometimes, you may want to resize or move a topic and have WhizNotes remember its position. For that, you should select this option so that the position of WhizFolder window along with any resized or moved topics will be remembered. As a result, next time those topics will always be opened in that position and size.

How can you recognize which topics have remembered this customized information? A double-headed arrow will appear to the left of a topic name in the list of topics. You can always "Reset any of these topics to their default sizes" by selecting them and selecting the menu item **Options-Reset selected topics to default sizes**.

Tip: If you want the topics opened with the caret positioned exactly at the last occupied position then you should select this option.

- **Open topics on top of previous topics:**

If you select this option, a new topic will always cover an existing open topic of that WhizFolder. This may be useful when you don't want to cover the list of topics by an open topic window. Using this feature you can easily simulate the **Pane** view provided by some Windows programs, if you like it.

New topic prefix:

If you want to make a WhizFolder as a sort of log or a journal then you can check either Date or Date/Time options with a choice of long or short date format. Once you do that and add a topic, a prefix will automatically appear with the current date/time for the new topic name.

Restore desktop when opening WhizFolder:

If you select this option then whenever a WhizFolder is opened, the previously open topics are opened automatically at the same position. This is useful for finding out where you left off last time.

Don't open topic window on add topic:

If you want to quickly build up a list of topic names without entering text in them then you can select this option.

Don't create backup file:

Normally, WhizNotes creates a .WBK backup file for each save to the WZN file. For large WZN files, this may waste space on your disk. You can check this option to avoid saving .WBK file for this WhizFolder. But, in that case, please ensure that you take regular backups of your WZN files so that you can restore them in case of a corruption.

Where are all the options kept?

For your information, the desktop options for a WhizFolder are kept in a file with the extension **WDK**. Whenever you do a **save WhizFolder**, the options are saved in the WZN file and the WDK file is deleted. This way, WDK files remain only till you make a change to the WhizFolder file. This avoids unnecessary save of complete WhizFolder files for any minor change in the options.

Please note that WDK files are for your environment only, and you can always delete them without losing any data; however, you may never want to delete them as they keep your latest unsaved options for a WZN file. Also, when delivering a WZN file to someone else, you need not give its WDK file along with it. Just give the WZN file but make sure that the latest options have been saved to it. How do you ensure that? For this, another menu option is provided: **Options-Prepare distributable WZN with latest options**. Use this to force saving of options to the WZN file whenever you have to deliver a WZN file to someone.

What is toolbar?

Frequently used menu operations are available on a toolbar below the menu bar. To know the operation associated with a button, just move the mouse cursor over it and a hint on the operation will be shown on the status bar at the bottom.

Also, when you press a toolbar button, its menu equivalent will be shown on the status bar. *If you don't want to execute the operation, move the cursor out of the button before releasing it.*

Show topics with no HelpTopic definitions

Shows those topics for which you have not yet defined HelpTopic statements.

It will select those topics for which the HelpTopic statements are missing.

It's recommended that you open a **~HelpAttributes** topic before performing this select operation so that you can drag and drop the selected topic names onto the **~HelpAttributes** topic. This will facilitate entering HelpTopic statements for these remaining topics.

Defining startup help macros

(only available in the registered version)

Two types of startup help macros are available in WhizNotes:

1. Startup help macros which execute when the help file is opened.
2. Startup help macros for a topic which execute when that topic is displayed.

Defining startup help macros for the help file:

You may want to execute some help macros when the help file is first opened. To do that, create a special topic with the name "~StartupHelpMacros" and put your help macros in this topic, each on a separate line. When generating help, WhizNotes will insert these macros in the **[CONFIG]** section of the help project (**HPJ**) file. Of course, you yourself are responsible for dealing with errors given by the help compiler for these macros.

Example:

Suppose, you want to add a "Close" button to close the help window. Usually such buttons are added by executing the help macro "CreateButton" at the startup. To define it as a startup macro, create a topic named "~**StartupHelpMacros**" and add the following macro to it:

```
CreateButton(`bt_close', `C&lose', `CloseWindow(`main`))'
```

This will make sure that the button is added as soon as the help is started. This way you can use any number of help macros as the startup macros, each on a separate line. Please refer to your Windows SDK help for more information on help macros and their syntax.

Defining startup help macros for a topic:

This can be done by the **Macros** option of the HelpTopic statement. For details, please see [Keywords, macros, and other characteristics of topics](#).

Also see:

[Inserting help macros](#)

Using 'Separator' topics to organize the topic list

Separators are special topic names beginning with characters "~~". These dummy topics allow you to divide the list of topics into separate visual sections. This allows for better visual organization of topics in the list. Separators are useful only if you are viewing the list in non-sorted order. Separators act as **comments** and are ignored for all the important processing, such as when doing help generation, exporting RTF or printing **all** the topics.

To add a separator to the list, use a topic name that begins with characters "~~". For example, you may use a name like "~~ ----- List of procedures A" before a set of related topics.

Note:

Although **separator** topics are not intended to contain any text, you can still enter text into them. That's why separators are **not ignored** for all the other operations, such as "Print selected topics" and "Export selected topics to text file". This is for your convenience, so that you can print them selectively or export/import them from text files.

Hints on using 'Convinc.Exe' utility with VB and Pascal

(only available in the registered version)

When generating help, you can select to generate an include file (.INC) containing the context IDs and context numbers for providing context sensitive help from your program. If your programming language is other than C, you can convert this file using the CONVINC.EXE utility. Here are the examples to convert the .INC file for use with Visual Basic and Turbo Pascal programming languages.

Converting .INC file for use with Visual Basic

Here are the steps:

1. Create a template file called **VBTMPL.TXT** and put the following line in it:
Global Const %context-string = %context-number
2. Assuming that your .INC file is called **helpids.inc**, enter the following command on the DOS prompt. Ensure that you have put the utility **convinc.exe** on your DOS path or in the current directory:
convinc @vbtmpl.txt helpids.inc vbhlpids.txt
3. When the command finishes, your VB format constants are available in output file **vbhlpids.txt** for use in your Visual Basic program. Now you can put this file in your VB project file (.MAK) so that the constants are available to your program. You can put the IDH names of the constants as values for HelpContextId attributes in various dialogs and menus.

Converting .INC file for use with Borland Pascal

Here are the steps:

1. Create a template file called **BPTMPL.TXT** and put the following line in it:
%context-string = %context-number;
2. Assuming that your .INC file is called **helpids.inc**, enter the following command on the DOS prompt. Ensure that you have put the utility **convinc.exe** on your DOS path or in the current directory:
convinc @bptmpl.txt helpids.inc bphlpids.txt
3. When the command finishes, your pascal format constants are available in output file **bphlpids.txt** for use in your pascal program. You will have to copy the contents of this file in the "Const" section of your pascal program.

How to help-compile in background

Do you know that in Windows 3.1 environment, you can control your DOS applications to run in a window and in background? You should take advantage of this feature so that you can switch to other applications when the help is being compiled.

This is simple to set up. Just start Windows PIF editor and open the file **_DEFAULT.PIF** which is in the Windows directory. Select the options: "Display usage: Windowed" and "Execution: Background". Then save the file. If you don't find **_DEFAULT.PIF** file then search for the keyword "Default PIF" in the PIF Editor's online help to see how to create it. If you face any memory shortage problems, please look at other options in the PIF Editor.

How to uninstall/move WhizNotes

Moving WhizNotes:

If you want to move WhizNotes to another directory, just create the new directory and copy all the files from the installed WhizNotes directory to the new directory and delete the original directory after carefully checking that you have indeed copied all the files to the new place.

Then run Setup program in the new directory and perform the operation 'Reinstall after a move'.

To move WhizNotes to another PC, the same method can be used.

Uninstalling WhizNotes:

Run the Setup program and perform the 'Uninstall' operation.

Terminology of help generation

Here is a list of some help generation terms to help you better understand the help generation features of WhizNotes:

Rtf-footnotes and hidden-text:

Details

If you were not using a Help authoring tool like WhizNotes, you'd need to constantly deal with RTF-footnotes and hidden-text. Almost all of the Windows help features are defined in an RTF document by inserting **RTF-footnotes**, marked with special characters and hidden text. Fortunately, with WhizNotes, you don't even have to know of the existence of RTF-footnotes and how they work. It's indeed a very high level help authoring tool.

Jump:

In Windows help, you can see some underlined phrases that allow you to go to a different topic window by clicking on them. These are called hypertext Jumps. For example, here is a jump: Window menu. WhizNotes allows you to insert jump statements by dragging topic names on top of the open topic windows. You can even test the link by double-clicking on such a jump statement. There are other variations of jump statements which allow you to show a different underlined text or even a bitmap as a hotspot.

Popup or a reference:

These are similar to jumps but are underlined with dashes. For example, here is a reference: Window menu. When you click on a reference, the indicated topic is opened as a popup window. This is suitable for small topics or definitions. In WhizNotes, the references are inserted similar to the Jump statements and the double-clicking also works like a Jump so that you can test the link. There are other variations of reference statements which allow you to show a different underlined text or even a bitmap as a hotspot.

Context string:

Details

Windows help internally uses special jump-identifiers for the help topics. Such identifiers are called **context strings**.

Contents:

In Windows help, at any time, you can press on a button **Contents** and a table of contents appears. WhizNotes generates such a Contents topic for you by default where it includes all the topics from your topic list. This is often not desirable and you should make your own topic with the name "Contents" where you can include only the top level topics with proper formatting for better readability. WhizNotes will use your topic "Contents" if it is present. Foreign users may want to designate a different topic for Contents. This can be done by using certain options called GlobalHelpOptions.

Keywords:

When you press the "Search" or "Index" button in Winhelp, you actually use a keyword search facility where a dialog box appears showing you a list of phrases or words. These are

keywords. On selecting a particular keyword, you are shown a list of topics related to that keyword. Such words or phrases are called **Keywords**. By default, WhizNotes assigns one keyword to each topic which is same as the topic name. If you don't want that, you can use HelpTopic definitions to avoid that. In addition, by using the HelpTopic definitions, you can define more keywords for a particular topic. Other statements to directly define keywords in the topic text are also available.

Browsing sequences:

Look at the top of this help window. You will see push buttons << and >> that guide you through a series of related topics. These are called **Browse buttons** and the groups or series of topics are called **Browse groups** or **Browse sequences**. WhizNotes allows you to form groups by simply a drag and drop of topic names to form special objects called ListBrowseGroup statements. It is very simple to manage all your browse groups with the ease of a drag & drop and free-format text editing to rearrange the topic names.

Help Macros:

Help Macros allow you to customize the Windows help to include more buttons or features like starting an external program. Startup help macros are those macros which execute whenever a help file is opened or a topic is displayed. WhizNotes allows you to define startup-macros for help file, startup-macros for topics as well as embedded macros in the text of topics. For details, search help for keyword **Help Macro**.

Conditional help generation:

This is a very powerful feature of WhizNotes that allows you to select topics and topic text for generating a different version of the help from the same WhizFolder file. Please do not confuse it with build-tags provided with the help compiler. WhizNotes' Conditional help generation is more powerful and more convenient to use than the build-tags.

HHML statements:

WhizNotes uses a special hypertext markup language, HHML, so that you can define hypertext links and other text attributes easily. HHML statements look like [**~jump...**], [**~bold...**], [**~startlist**]. WhizNotes allows you to make your custom HHML statements to add to WhizNotes' formatting and hypertext capabilities.

Help compilation:

When you select "Create Help" operation, WhizNotes generates the following files which have the same file name as the WZN file but different file extensions.

- An **RTF** file that contains all the text in your topics in the form of help RTF statements.
- An **HPJ** file that includes the above RTF file and also describes other options as selected by you.
- A **BAT** file that contains the command to execute the help compiler on the above **HPJ** file.
- If you selected option "Also generate the include file containing context IDs" then it also generates an **INC** file containing the context numbers for all the context strings. You can find out context strings assigned to your topics from this file.

Then it runs the BAT file to perform the help compilation. If the compilation succeeds, an HLP file is created by the compiler. After the compilation, it preserves the files RTF, HPJ and BAT.

So that, if you want, you may change RTF file or HPJ file in another editor or help authoring system.

Help Generation Dialog Box

This dialog box appears when you select the menu **Tools-Create Help...** to generate Windows hypertext help from a WhizFolder file. Here's a list of options available on the help generation dialog box. You may see details on any option by clicking on it. To come back to this list, press the 'Back' button.

- Title for Help Window
- Copyright Notice
- Type of font
- Options
 - Compress Help file
 - Optimize for CD ROM
 - Remove browse buttons
 - Don't overwrite HPJ
 - No Wait for Compilation
- Advanced Options
 - Selectively generate by using tags
 - Just create RTF files...
 - Just run the help compiler
 - Also generate Include file containing Context IDs

Here's a detailed description of the options:

Title for Help Window:

The title that you enter here will appear as the title of the generated Windows Help window.

Copyright Notice:

You can put your Copyright Notice here which will appear in the about box of the generated Help. For example, please look at the About Box of this help that you are reading.

Please note that you can enter a Copyright symbol by typing 0169 on the numeric keypad while keeping the ALT key pressed.

WARNING: However, if you enter other special symbols, they might not appear properly in the generated help's about box. To fix that, you'll have to exit Windows 3.1 and run its setup to install the multilingual font instead of the regular OEM font. WhizNotes processes only the Copyright character with a special work around to avoid this problem.

Type of font and Tabsize:

For Windows 3.1 help authoring, Microsoft documentation recommends that you use "MS Sans Serif", size 10, font for your help documentation. Also, the default tab distance is 0.5" which is equivalent to tabsize 6 in the above font. Microsoft help also follows the same standard in Windows 3.1. If you want another font as default font or a different tabsize, select the "Same as WhizFolder" option. In this case, the font and tabsize will be same as the font and tabsize for the WhizFolder, one that you specified in the "Desktop options" for the WhizFolder. For Windows 95 recommendations, see the discussion later in this topic.

Note that the tabsize that you select here affects the indents used in the StartList and Indent statements. If you don't like that, specify a ListIndentInches value in the [GlobalHelpOptions](#) statement. Also, if the TabSize option on the WhizFolder can't give you the exact tab positioning that you want, use an option TabStopsInches from the [GlobalHelpOptions](#) statement to specify a tabsize in inches for your help.

Options:

Compress Help file:

With this option, you can generate a compressed Help file that requires less space but needs more time for creation. It is best to use this option when you are finally generating a help file to be distributed to your customers.

Optimize for CD ROM:

With this option, the help file is optimized so that it is read more efficiently from a CD ROM. See your help compiler documentation for more details.

Remove browse buttons:

The browse buttons are identified by << and >> symbols on the help window. They are supposed to guide the user through a series of related topics. You should remove these buttons if you have not made special efforts to define the browsing sequence of topics as described in [Browsing sequences of topics](#).

Don't overwrite HPJ:

This option is useful if you have manually edited the HPJ file by a text editor or some other tool. For example, with the new HCW compiler, you may want to edit the HPJ file for using the new features available with HCW. In that case, you don't want your changes to be overwritten by WhizNotes. If you select this option, WhizNotes won't overwrite the HPJ file. However, it is recommended that you don't use this option. If you do then any changes you make to some options in this dialog will be ignored as the new HPJ file won't be generated for those changes. This behavior may confuse you unless you really understand the consequences of using this option.

It's better to use the HPJ customization options of WhizNotes to generate an HPJ file the way you want it to be. It's not difficult to change existing sections or to add new sections in the HPJ file. For details, see [Customizing Help Project \(HPJ\) file](#). But, if you must edit the HPJ file in HCW, it's best to copy the HPJ file created by WhizNotes by another name and always use that with HCW for compiling your help. Once, you're satisfied with the changes, it's better to merge them in the HPJ file generated by WhizNotes by using the above options.

No Wait for Compilation:

With the new Windows based HCW compiler, WhizNotes automatically selects this option. The reason is that after the compilation completes, HCW doesn't exit and there's no signal for WhizNotes to know when the compilation finishes.

With old DOS based help compilers, you can select this option so that you can work on other WZN files if the compilation takes too long. This way, you can start multiple compilations for different long WZN files simultaneously.

Advanced Options:

To use the advanced options, you first need to understand what files are created by WhizNotes during the help generation process. For your information, the files with following extensions may be created by WhizNotes. Some of them may or may not be created depending on certain conditions. All these files have the same name as your WhizFolder file, but have different extensions:

- **RTF** All your help topics in Rich Text Format (RTF), always created
- **HPJ** Help Project File, always created unless disabled
- **BAT** Batch file to run help compilation (DOS based compiler only)
- **HPO** Output file for help compilation (DOS based compiler only)

Just create RTF files...:

This operation was useful in the earlier versions for stopping just before the compilation so that you could edit the RTF file to change certain text formatting with your favorite word processor. *But now, WhizNotes does not delete the RTF and HPJ files after a successful compilation so that they are always available for modification. Hence, this option is no longer necessary for that purpose.*

However, this option may still be useful with HCW if you're using another HPJ file for compilation. In that case, keep this option selected. Then, for any change in your WZN file, you can regenerate the RTF file and then switch to HCW for compiling the other HPJ file.

Just run the help compiler:

This operation was useful in the earlier versions for the same reasons as mentioned above. Once you make your custom changes to the **RTF** or **HPJ** files as mentioned above, you need not manually start the help compilation. You can just check this option and WhizNotes will only run the help compiler without disturbing the changed **RTF** or **HPJ** file.

Of course, if you were to uncheck this option later, remember that next help generation will regenerate the RTF and other files and will overwrite your changes.

Note that if you use this option, the help compiler forced through the `forceHelpCompilerToUse` option of the `GlobalHelpOptions` is not used. Instead, the help compiler associated with the HPJ extension is used.

Generate Include file containing Context IDs:

This option is for the programmers who want to integrate calls to the help topics in their program. A C language header file is generated that contains the `#defines` for the Context strings used by WhizNotes. For details, please see [Features for context sensitive help](#).

Selectively generate by using tags:

This needs a fairly advanced discussion in a separate topic. For details, please look at [Conditional help generation](#).

Sample WZN file to demonstrate Help Generation

If you have already compiled the SAMPLHLP.WZN file as given in the introduction then you may skip this topic.

If you think, you have a help compiler installed on your system, just open the sample folder **SAMPLHLP.WZN** with the menu **Open-WhizFolder...** Then, click on the menu **Tools-Create Help** to see WhizNotes in action. You'll see the topics in the sample folder converted to a Help file in a matter of seconds. If the help compiler is not installed properly, you'll get a suitable error message. In that case, please consult the topic [Setting up the Help Compilation environment](#).

Now that you've seen a demo, open a few topics in the sample folder by double-clicking on them. They're quite easy to understand, and if something is not clear, you can always search the online help for more details. I'm sure you'd like to make a few changes in the text and try generating help again. By all means, do it! You can even try [inserting more hypertext jumps](#) between topics. It's as easy as a drag & drop. Please see [More about the demo](#) for more details on this demo WZN file.

You're beginning to get an idea of how easy it is to create hypertext help with WhizNotes. What you've seen are only a few of the basic help capabilities. Want to know complete details on what WhizNotes has to offer for help generation? Please read [Help generation with WhizNotes](#).

Making your text bold, italic, etc.

There are HHML statements to make the text bold, italic etc. There are two ways to use these statements:

1. Apply attribute to enclosed text only. For example, a statement [**~BOLD text lines**] will apply the attribute bold to the enclosed text "text lines". The enclosed text can be as large as a paragraph.
2. Another way is to turn on the attribute till the end of the para or a turn-off whichever occurs earlier. For example, [**~bold**] will turn on the bold attribute till it encounters the end of the para or a [**~wResetFormat**] statement.

The following attributes are available for marking the text as shown above:

- ~bold
- ~italic
- ~wBoldItalic
- ~wSmallCaps
- ~wStrikeOut
- ~wUnderline - *It's strongly suggested that you don't use underlined text as it would cause confusion with the underlined hypertext links on a monochrome screen.*

What if you want bold and underlined? You can do that if you know RTF. You need to create your own HHML statements by specifying a powerful WhizNotes feature called RtfMacros. For details, see [Creating your own custom statements with RtfMacros](#).

Specifying jumps & references to other help files

To specify jumps and references to other help files, use the following statements:

```
[~JUMPTOCONTEXT {VisibleText = "visible-jump-text", ContextString = "context-string",  
HelpFile = "help-file-name"}]
```

```
[~REFTOCONTEXT {VisibleText = "visible-jump-text", ContextString = "context-string",  
HelpFile = "help-file-name"}]
```

Explanation of parameters:

- **VisibleText** - The text to appear underlined
- **ContextString** - Context string of the topic to jump to
- **HelpFile** - (optional) name of the help file containing the above topic

Integrating other RTF files:

You must have noticed that if you don't specify a **helpFile** then you can jump to a topic in the same help file. With WhizNotes generated help, you may rarely use this feature to jump to a topic in the same help file. But there are instances where it can be useful.

See the sample WZN folder for an example of how this statement can be used to jump to the middle of a topic or to another place in the same topic.

Let me give you another example of when it would be useful. If you generated some of the RTF code using some other help generator (for instance, using the context sensitive RTF help generated by Visual C++ or Borland C++) then you can integrate these RTF files into your own help by including them in **HPJ** file in the section **[FILES]**. See [Features for context sensitive help](#) for how to do that.

Once you do that, you can use the above statements to jump to the topics defined in these RTF files.

Jumps between WhizFolders

You must have noticed that if you drag and drop a topic name from one WhizFolder to a topic of another WhizFolder, it gets inserted as a **JumpWzn** statement so that you can double-click on this statement and jump to the other WhizFolder. Currently, JumpWzn statements are only provided for hypertext organizer use. They can't be processed when generating help and cause an error message.

Inserting direct RTF code

This option is for advanced help-generation users. If you are familiar with Rich Text Format, you can insert direct RTF code in your topic text by using the following statement:

[~RTF your-RTF-code]

If you have a big block of RTF-code consisting of multiple paragraphs then you can enclose the block between the lines containing **[~RtfOn]** and **[~RtfOff]** statements.

Once you use your own RTF code, chances are that you will get help compiler errors. Help compiler usually gives hex locations in the error messages. In such a case, you can use my **WZLOCATE.EXE** DOS utility program (available only with the registered version) to locate a hex location within an RTF file.

Also see:

[Creating your own custom statements with RtfMacros](#)

Inserting help macros

(only available in the registered version)

Have you ever looked at the topic **Help macros** in your Windows SDK help? This is a powerful feature of Windows Help that lets you customize your help more like an application. WhizNotes allows you to easily insert help macros in your text by using the statement:

```
[~HELPMACRO {visibleText = "visible-text", macro = "macro-statements"}]
```

Explanation of parameters:

- **visibleText** - The link text to appear underlined
- **macro** - Help macro statements

When the user clicks on such a link text in the generated Windows help, the specified help macros will be executed.

Please note that since WhizNotes uses double-quotes to enclose the macro-statements, it is easier for you to use single-quote pairs in macro arguments. Single-quote pairs consist of **left-quote (`)** and **right-quote (')** present on two different keys on the keyboard and their order is important. If instead, you want to use double-quotes in macros, you will have to repeat each double-quote twice for WhizNotes to evaluate them properly. Hence, it is more convenient to use single-quote pairs.

Example:

Try this macro in your topic text:

```
[~helpmacro {visibleText = "A clock program", macro = "ExecProgram(`clock.exe',0)"}]
```

Here is how it will look in the generated help: A clock program. Click on it and it will start the clock program.

Also see:

Defining startup help macros

Keywords, macros, and other characteristics of topics

Conditional help generation

This is one of the most useful features of WhizNotes. Do not confuse it with the Build Tags as specified in the Help Compiler documentation. Rather, WhizNotes conditional help generation is even more powerful than the one provided with the compiler. Here is a discussion on when to use it.

Sometimes, you may want to use the same WhizFolder for generating various levels of help. For example, for a demonstration program you may need a little different text as compared to the registered version. Provided that most of the text is similar, how can you control which text in a topic is to be used and which topics are to be skipped? WhizNotes conditional help generation can do it for you. But you must understand that if you make logical errors in specifying condition statements, WhizNotes can not always catch your mistakes. Hence, you are responsible for making sure that the statements for conditional help generation are logically correct.

There are three aspects of conditional compilation:

1. Specifying one or more **TAGs** (identifiers) in WhizNotes Help Generation Dialog to specify what all combinations you want.

Examples of tags are "Demo", "Registered", etc. When entering more than one tag in the dialog, please use semicolons to separate them.

2. Preparing topic text with the following statements to control what text gets included and what text gets excluded when a tag is present or not present. These statements are **[~IFTAG..], [~IFNOTTAG..], [~ELSE],** and **[~ENDIF]**. C programmers should understand that this is similar to the C directives **#ifdef, #ifndef,** etc. with one exception: WhizNotes also allows them to be embedded anywhere in the paragraph text.
3. If whole topics are to be skipped depending on whether a tag is defined or not defined then you may use similar named keywords when defining help attributes for topics. For details, please see [Keywords, macros, and other characteristics of topics](#).

Examples:

I will give you two examples of using conditional tags:

1. Suppose you want a line in your topic to read "This production version gives you the following additional facilities:" in the help for production version; and "This demo version restricts use of these features:" in the demo version of your program's help.

Here is how you would do this:

- Decide to use a tag "PRODUCTION".
- Prepare your text like this:

```
[~IFTAG PRODUCTION]
This production version gives you the following additional facilities:
[~ELSE]
This demo version restricts use of these features:
[~ENDIF]
```

- Now if you define the tag PRODUCTION in WhizNotes Help Generation Dialog and

then generate help, you will get the first line; whereas, if you remove the tag from the dialog and generate help, you will get the second line. The use of ELSE part is optional. You could have used an **[~ENDIF]** there if you did not want any line to appear for the demo version. Similarly there is an **[~IFNOTTAG...]** statement that may be useful for negative conditions.

2. You just want to change the word "production" to "demo" in the above example of the first line. To do that, use the statements in this manner:

This `[~IFTAG PRODUCTION]production[~ELSE]demo[~ENDIF]` version gives you the following additional facilities:

As I said earlier, this is a powerful feature that may save you lot of rework to maintain different help files for different versions.

Also see:

[Keywords, macros, and other characteristics of topics](#)

Browsing sequences of topics

(only available in the registered version)

What is a browsing sequence? If you look at the ribbon bar of this help window, you will notice two buttons << and >>. Try to go in forward direction by pressing the button >>. You will see that on each press, it guides you through a series of related topics and when you reach the end of the group, the button is disabled. You can no longer proceed in the forward direction, you have to press the button << to move back through the topics. Of course, as a help author, it is I who decided what topics are related to each other and then placed them in separate groups in a sequence. How did I do that? I used WhizNotes' **ListBrowseGroup** statements to specify these groups.

WhizNotes makes it very simple to assign browse groups. All you have to do is, make a special topic named **~HelpAttributes...** and define the browse groups using the statement syntax given below. If one topic is not enough, you can define more of these topics, provided their names begin with the characters **~HelpAttributes**.

Note that the **ListBrowseGroup** statement is not an HHML statement, but is [An HHML Tool](#).

Format of a ListBrowseGroup statement:

```
ListBrowseGroup
{
    "topic-name1",
    "topic-name2",
    ...
}
```

You can code the above statement in one line or multiple lines in free format. Also, you can define as many browse groups as you want. But remember, in Windows help, a topic CAN NOT be present in more than one browse groups.

To avoid typing all the topic names, WhizNotes allows you to drag multiple topic names and drop them on your **~HelpAttributes** topic window. All you have to do is to enter the beginning and end of the ListBrowseGroup statement as shown above. After dropping the names, you may have to rearrange them if you want a different sequence. If you have formatted your "Contents" topic in a hierarchical manner then it is best to create browse groups based on your contents.

Special topic **~HelpAttributes** also allows you to define other options for topics and for controlling help generation. For more details, see [Keywords, macros, and other characteristics of topics](#) and [Global options to control help generation in general](#).

Keywords, macros, and other characteristics of topics

WhizNotes allows you to specify keywords, help macros and other options of topics for controlling the help generation. Such options need to be defined in a special topic named **~HelpAttributes**. You need to specify topic options using **HelpTopic** statements as described here. If one topic is not enough, you can define more of these topics, provided their names begin with the characters **~HelpAttributes**.

Note that you can quickly locate the HelpTopic statement for a selected topic by a right click on the topic list. Note that the **HelpTopic** statement is not an HHML statement, but is [An HHML Tool](#).

Note that it isn't necessary to define **HelpTopic** statements for all your topics. But if you are a serious help author, you would certainly want multiple keywords for a topic and a HelpTopic statement is the best way to do that. If you really want to have total control on all your topics with a HelpTopic statement, then after adding multiple topics, you may want to find out what topics do not have a **HelpTopic** statements defined. To do that, use the menu item **Tools-Show topics with no HelpTopic definitions**.

Format of a HelpTopic statement:

```
HelpTopic "topic-name"
{
    Keywords = "",
    TopicNameIsNotAKeyword = "NO",
    Macros = "", (only available in the registered version)
    UseNonScrollingTitle = "NO",
    VisibleJumpText = "",
    SkipIfTag = "",
    SkipIfNotTag = "",
    ContextString = "",
    ContextNumber = ""
    NoTopicTitleAtTop = "NO",
}
```

Tip: [A note on quoted values](#)

Keywords:

Keywords are words or phrases that you specify in the "Search" or "Index" window of the generated help. By using keywords, a user can get a list of topics related to that keyword. In WhizNotes, by default, the topic name itself is a keyword for a topic. If you want, you can add more keywords by assigning them to this attribute. Make sure that if you are defining more than one keywords for a topic, you must separate them with a semicolon.

You can also define keywords in the text for a topic. For details, see [Defining keywords for topics](#). However, in many cases it's much easier to define keywords in the HelpTopic statement.

TopicNameIsNotAKeyword:

As already seen, WhizNotes generates a keyword same as the topic name, that is, it uses the default value "NO" shown here. Make it "YES" if you do not desire this for a topic. When deciding on this option for a topic, just think for a moment. Is the user of your help

likely to type first letter of the topic name to look for this keyword? If yes, let the name be a keyword. If not, remove the name from the keyword list. For example, if you have a topic named "What's new". Naturally, user is more likely to type 'N' to look for new features. Hence, you should make **TopicNameIsNotAKeyword = "Y"** and define the keyword "New features" in the **keywords** option.

Macros:
(only available in the registered version)

You can define startup help macros for a topic with this attribute. They will be executed whenever this topic is seen by the user. A detailed discussion of Help Macros is outside the scope of WhizNotes. You may look at your SDK reference for details on their syntax and usage. For example, you should know that startup macros will not work for a referenced or popup topic. **Please note that since WhizNotes uses double-quotes to enclose the whole macro string, it is easier for you to use single-quote pairs to specify options in your macros.** Single-quote pairs consist of **left-quote (`)** and **right-quote (`)** present on two different keys on the keyboard and their order is important. If instead, you want to use double-quotes in macros, you will have to repeat them for WhizNotes to evaluate them properly. Hence, it is more convenient to use single-quote pairs.

UseNonScrollingTitle:

This is a very attractive feature of Windows help. For those topics, which you don't plan to use as popup or referenced topics, you can specify "Y" for this attribute. The result will be an attractive non-scrollable heading in the generated help. For an example, look at the heading of this topic. You can even control the background color for the non-scrolling region by editing the HPJ file. Please see [Displaying topic title in a non-scrolling region](#)

VisibleJumpText:

Don't you wish sometimes that for some jumps you want to present a different visible text to the user than the original topic name? You can do it in two ways. Normally, you may use [JumpShowText](#) statement. But, for a particular topic, if you always want to use a different visible text then use **VisibleJumpText** attribute. In this case, the text that you specify will be used like an Alias for the topic name everywhere--in the jumps and ref statements to that topic, in the topic title and in the topic name keyword.

You may ask, "If everywhere the alias is used, why do I need the original topic name?" Well, as I have said again and again, the users use WhizNotes in their own clever ways. Currently there is no facility in WhizNotes for organizing the topic list in levels (hierarchy). One smart user wanted to give topic names like "I_0.Who.There" and "I_0.Who.Here" to simulate hierarchical levels of topics. But then, he didn't want such topic names to be actually used in the text of the generated help. Such a problem is solved by the above feature.

SkipIfTag and SkipIfNotTag:

These options allow you to completely skip a topic depending on the presence or absence of a tag. For more details, please see [Conditional help generation](#).

ContextString:

WhizNotes generates [Context strings](#) based on the topic names, automatically. So if you are using them to provide context sensitive help for your application, you need not worry

about changing your program when you add or rearrange your topics. But if you want, you can still override the default Context string by defining a value for this attribute. **The value should be up to 255 chars long and should only contain 'ASCII' alphanumeric chars or '_' char. WhizNotes forces this rule to avoid invalid identifiers in the C header file so as not to choke the C compiler.**

ContextNumber:

This feature is for the programmers who are generating context sensitive help. This option lets you override the contextNumber assigned to this topic in the generated Include file. For more details, please see Features for context sensitive help.

NoTopicTitleAtTop:

Sometimes, you want to directly code a topic title in the topic in the format you want. For example, when it's a popup topic defining a term, you may not want to show a title. In such cases, you can make **NoTopicTitleAtTop = "Y"** so that HelpHikes puts no topic title at the top.

Also see:

A note on quoted values
Defining startup help macros
Inserting help macros
GlobalHelpOptions

Creating your own custom statements with RtfMacros

(only available in the registered version)

If you know RTF syntax then after using [**~BOLD...**], [**~ITALIC...**], and similar HHML statements, you may have wished that you could create some simple statements of your own. However, if you try to code RTF in a topic text then there is no way for WhizNotes to know whether you really intend it to be interpreted as RTF or not. Consequently, it processes it as a literal text. You can insert direct RTF code in your topic text using the statement [**~RTF...**]. In this case, WhizNotes will not touch the enclosed text and will write it verbatim to the RTF file.

But, there is an even more powerful feature. You can define your own statements in a special topic with the name: **~RtfMacros**. If one topic is not enough, you can define more of these topics, provided their names begin with the characters **~RtfMacros**.

Note that the **RtfMacro** statement is not an HHML statement, but is An HHML Tool.

Format of RtfMacro statement:

```
RtfMacro macro-name
{
    RTFBeforeText = "",
    RTFAfterText = "",
    RtfIfNoText = "",
    InsertBlankBeforeText = "YES"
}
```

Tip: A note on quoted values

Explanation of how RtfMacros work:

RtfBeforeText and RtfAfterText:

Suppose you defined an RtfMacro called **MyMacro**. When a statement [**~MyMacro some text**] is encountered in your help text then WhizNotes macro substitution takes over. First, it looks at the statement to see if some text is enclosed. If there is text, it processes **RtfBeforeText** defined for that macro and writes it to the RTF file. Next, it writes the enclosed text to the RTF file, taking care to properly put non-ASCII characters as per the RTF rules. Next, it processes the **RtfAfterText** defined for the macro and writes it to the RTF file. It's usually necessary that you code a '{' at the beginning of **RtfBeforeText** and a '}' at the end of **RtfAfterText**.

RtfIfNoText:

If you use the statement without an enclosed text: [**~MyMacro**], then only **RTFifNoText** is processed. Hence, you should not use any '{' or '}' characters in the RTF for this parameter so that it can work as a flag.

InsertBlankBeforeText:

By default, a blank is inserted between your **RtfBeforeText** and the enclosed text in the statement. You can set this parameter to "NO" to avoid that. This is useful for making those RTF statements where a given value is concatenated to an RTF symbol without an intervening space character. Example: \sl288.

Example:

Let us take a concrete example. Suppose you want to create a new statement called **[~BoldUnder...]** that will make the enclosed text both bold and underline. How do you do that?

Here are the steps to do this:

1. Create a topic with a name that starts with **~RtfMacros**.
2. Define the macro in this topic as follows:

```
RtfMacro BoldUnder
{
    RTFBeforeText = "{\b\u|",
    RTFAfterText = "}",
    RtfIfNoText = "\b\u| "
}
```

3. Now try using the statement **[~BoldUnder...]** in the text of one of your topics. When you generate the help, you will see that your custom **BoldUnder** statement works! Also try using just **[~BoldUnder]** without enclosed text. It will work like a flag making rest of the paragraph bold-italic or till it finds a **[~ResetFormat]** statement. This is due to the **RtfIfNoText** parameter.
4. You can make more such HHML statements with RtfMacros and extend the HHML language. But remember, now you are entirely responsible for coding correct RTF in your definition. You will have to understand any Help Compiler errors generated by your RTF code yourself. The help compiler messages usually indicate the location of an error by a hex location. For your convenience, I have created a small DOS based utility to help you locate any hex locations in the RTF file quickly. It is **WZLOCATE.EXE**, available only with the registered version. Just run it under DOS without any parameters and it will show you the syntax to use it.

Also see:

[A note on quoted values](#)

More about the demo

Understanding the SAMPLHLP.WZN folder:

You should read the topics "Read me first" and "Some basics" in the sample WZN folder. They will guide you on how to go about knowing more about help generation features used in the sample folder.

It's best to open the "Contents" topic and read all the topics one by one by double-clicking on them. You should also compare the topic text with the same topic generated in Windows help to see how it looks.

Please don't ignore the special "~HelpAttributes" topics in SAMPLHLP.WZN. These are special topics that control how the help is generated for particular topics.

Don't forget to look for more such sample folders:

In the registered version, you may find more sample folders demonstrating help generation. The exact names are not given here as more of them will be added depending on the needs of the users.

Help generation with WhizNotes

Here, we'll see what smart help generation features WhizNotes has to offer. After reading this topic, you'll also be in a better position to judge a help authoring tool and to know what you should look for in any hypertext authoring system. If, so far, you have compared help authoring tools based on their WYSIWYG capability, then you should reconsider your preferences. I'm not saying that WYSIWYG is undesirable. But if you get a WYSIWYG tool which does not deal properly with the other features of Windows help then it's not so useful. In authoring, the text presentation comes last. There are many many other hurdles to cross which are more difficult and tiring than the text presentation and beautification. Read this topic and find out for yourself.

I'll assume that you already know the features available in Windows help and understand the terminology, such as, a **browsing sequence**, a **keyword** or an **RTF-footnote**. If not, you should first read the topic [Terminology of help generation](#).

What you should look for in a hypertext Help Authoring tool:

- **Multiple window authoring**

Details

Windows help presents the online help to the user primarily in a single window. But when it comes to authoring, would you like to author in only one window?

- **Direct access to a help topic**

Details

Serious online help often involves hundreds of topics. You should be able to quickly get to a topic.

- **Management of context strings**



This is a big task of Windows help authoring which many of the help authoring tools solve only superficially.

- **Ease of inserting hypertext links**



A hypertext link generally jumps to another topic. But it can also display the other topic in a popup window or in a secondary window, may cause a jump to another help file, may execute another program by a help macro and so on. With many help authoring tools, you need to know the context string of the topic you are jumping to. Not so with WhizNotes.

- **Management of keywords, startup-help-macros, etc**



With many help authoring tools, at most you might get a dialog to enter keyword-RTF-footnotes and macro-RTF-footnotes in the topic text. It is very inconvenient in a word-processor if you need to compare keywords of two different topics by looking at their

footnotes. Not so with WhizNotes.

- **Managing browsing groups and sequences of topics**



This is another task which should be handled in a logical manner. You should be able to make one or more groups of topics so that a person reading the help for a group can be guided by the browse buttons through all the topics in that group.

- **Conditional compilation**



WhizNotes has its own conditional compilation features which are superior to the **build tags** provided by the help compiler. Not only does WhizNotes allow you to combine separate topics for a compilation, it also allows you to mark text-blocks in your topics which are present, not present or are different in a particular version of help. Very useful for maintaining a demo and a production version of online help from the same WhizFolder.

- **Easy to understand hypertext markup language**



WhizNotes gives you a special hypertext markup language (we call it **HHML**). If you know the terms **world-wide-web** and **HTML**, you already have an idea of how hypertext markup languages work. WhizNotes also gives you a smart customizable popup menu to conveniently enter HHML statements.

- **Error handling**

A help authoring tool should be able to catch hypertext-link errors when creating the RTF for help compilation. If it doesn't, you're likely to see a huge error listing from your help compiler which is difficult to resolve. WhizNotes catches most of your errors when creating RTF and opens your topics at the proper position so that you can quickly correct the error and proceed. By using smart-build along with this smart error positioning, your help generation will be about as painless as it can be.

- **Customization features**



Whatever help authoring tool you use, you'll always want to customize some features to your liking. WhizNotes provides a number of customization features ranging from simple to complex.

- **Features for programmers**



If you're generating online help to be used by a program, you'd like to get a header file containing all the context numbers of the generated context strings. Also, if you're using a language other than C, you'd like to convert this header file to your own programming language. WhizNotes has all those features and it also supports overriding the default context numbers and their range.

More features are available in HelpHikes Pro

If you want more features, you should get [HelpHikes Pro](#), a more advanced Help Authoring System than WhizNotes. [HelpHikes Pro](#) will continue to evolve, and any new help features will be added only in [HelpHikes Pro](#). For details, please see [HelpHikes Pro, additional features](#).

Customization features

WhizNotes gives the following categories of customization:

- There are [HHML statements](#) to customize many features of generated help.
- You can add your own HHML statements by using [RtfMacros](#).

The real power of RtfMacros feature is evident in [HelpHikes Pro](#), where you can use other RtfMacros in the definition of an RtfMacro recursively. This way you can even define text styles in HelpHikes Pro.

- If you find that you must use a WYSIWYG editor to add certain text formatting features which aren't possible with HHML, then you can edit the intermediate RTF file by the WYSIWYG editor to put in the changes and run the BAT file to compile the RTF file.
- If you need to change the HPJ file for any reason, you can add special topics in your folder that can replace, can add to or can create a new HPJ section in the project file. In addition, you can also specify an external HPJ file to be used for help compilation. See [Customizing Help Project \(HPJ\) file](#) for more details.
- The popup menu feature **WhizMenu**, that allows you to enter HHML statements, can be customized to add more HHML statements that you create or want to change entry of. See [WhizMenu for quick entry of canned text and HHML statements](#) for more details.
- For the new Windows based help compiler, WhizNotes selects switch '\C and \R' by default so that your HPJ file is compiled and the help file invoked automatically for testing. If you want to introduce additional switches, you can do that in the INI file. See [Less often used INI flags](#) for details.
- You may want to reuse substantial portion of your help text for the printed manual. For this, please don't use the RTF file which is generated for help compilation. Instead use the menu **Tools-Export non-help RTF** to export a clean looking RTF file which doesn't contain hypertext related footnotes.
- Foreign language users get features to ensure that correct RTF or OEM character set is generated in their RTF and HPJ files. In most cases, this is done automatically by WhizNotes. They can also specify a different Contents topic name by [GlobalHelpOptions](#).

Multiple window authoring

In hypertext authoring, or in any writing activity, it helps a lot if you are able to view many of your related or unrelated topics side by side. While working on one help topic, you may suddenly feel a need to add or change another topic. Once you use the multiple-window environment of WhizNotes, you'll never go back to single window authoring. Multiple window authoring helps you brainstorm your ideas. It compliments the random thought processes that occur while researching or writing on any subject.

What's more, WhizNotes contains sophisticated features to remember open positions of topic windows. Use them to your advantage.

Direct access to a help topic

In WhizNotes, you always work from the folder window which contains a list of the help topics. There are many-many features to rearrange, sort and search for the topics. To open a desired topic, you just need to double-click on it. Or, you can select multiple topics and press **Enter** to open all of them at once. WhizNotes also contains features to remember last open positions of topics and restore a complete desktop image of the open topics.

In the past, when writing, you might have often wished for a big desktop space where you can rearrange your open notes, look at them to get the big picture and open them automatically at the same positions next day. Well, physically this may not be possible but WhizNotes makes your computer do that for you.

Management of context strings

Superficial help authoring tools will only give you a dialog box to enable entry of the context-RTF-footnote in the topic. But it's you who must think up and assign context string to the topic and it's you who must remember it when adding a jump-RTF-footnote for a hypertext link to this topic across your whole document. This can be a nightmare.

What we need is a total freedom from the management of context strings. WhizNotes does everything for you transparently. It assigns and manages context strings for you based on your topic names. In fact, many WhizNotes users may not even know the existence of context strings. It works like magic. Also, if ever a need arises for you to force the context string of a topic to a string of your choice, you can do that too. No restrictions.

Ease of inserting hypertext links

With WhizNotes, the most used jump and popup links can be inserted by just a drag and drop of the topic name from the folder on to an open topic window. Or, you can copy a jump statement to the clipboard from the folder or from an open topic window and paste it into a topic. For other types of links, WhizNotes provides its own logical statements which can be quickly typed or inserted in the topic text using the right-click WhizMenu.

Management of keywords, startup-help-macros, etc

WhizNotes gives you a [**~wKeywordsHere ...**] statement to insert arbitrary keywords in the topic text.

WhizNotes gives you another more logical way where you can keep all the keywords or macros of the related topics close to each other. This is done in an object oriented manner by treating a topic as a **HelpTopic** object and treating the keywords and macros as attributes of that object. You simply code a HelpTopic statement for a topic and assign the **keywords, macros** and other attributes to it. Even if you don't assign, there are convenient defaults. For example, WhizNotes always defines one keyword for the topic which is same as the topic name. You keep the HelpTopic statements for a group of topics together, in a special separate topic. There are also features to quickly locate a HelpTopic statement for a topic if you want random access to the attributes.

A HelpTopic statement also lets you assign other useful options of a topic, all in one place and in free-format text.

Managing browsing groups and sequences of topics

A superficial tool may only help you to enter a browse-RTF-footnote with a dialog. It's you who must assign and keep track of group and sequence IDs. The process is so tiresome that many authors tend not to use this useful aspect of help authoring.

WhizNotes allows you to form groups by simply a drag and drop of the topic names to form special objects called ListBrowseGroup statements. It is very simple to manage all your browse groups with the ease of a drag and drop and the free-format text editing to rearrange the topic names.

HelpHikes Markup Language (HHML)

What is HHML?

Here's an example of an HHML statement: to make a text-block bold, you select the text-block, press right menu button and click **Bold** on the popup menu that appears. This encloses the selected text in a [**~bold ...**] HHML statement. Instead of using the menu, you could also type in the statement. No restrictions.

When you generate help, the text appears as bold in your generated help. Similarly, there are HHML statements to make the text italic; to insert a bitmap in your text; to insert nested bulleted or numeric lists, and so on.

There are several advantages of using such markup languages. If you're interested, please take a look at Hypertext markup languages.

Hypertext markup languages

Advantages of hypertext markup languages:

1. They simply provide more powerful hypertext features in the form of logical high level statements. For example, consider the problem of adding a hypertext jump in an RTF editor like Microsoft word. See [Rtf-footnotes](#) for details. If you go to all that trouble, you're getting all the headaches of hypertext, instead of getting its advantages.

In WhizNotes, the jump statement appears as [**~jump topic-name**]. It is easy to see which topic it's jumping to. You can even test the jump by double-clicking on it. And to change to another topic-name, just type in the name of that topic or even easier, change it by a drag and drop or by a copy operation via clipboard. Moreover, you can quickly change the behavior of a jump statement to other [HHML](#) statements like **JumpShowText** or **JumpShowBitmap** by quickly typing in the changes.

2. Note that some help authoring tools might try to fool you by bragging about a GUI interface. A GUI interface does not necessarily make the tasks easier if it's superficial. If you're not convinced, try this: try to create the actual table of contents in a .CNT file using the HCW compiler, which gives a GUI interface with some nice looking features. But when you go to use it, you will find that it expects you to remember and enter context strings (topic IDs) for each content entry. Very tiring indeed! Whereas, with [HelpHikes Pro](#), you can code high level HHML statements to achieve the same results. In that case, you don't have to worry about the context strings and can work with just a drag and drop of topic names.
3. You get platform independent hypertext authoring. For example, consider the HTML used for web (WWW) authoring. HTML documents can be viewed on any browser whether it is on a unix or Windows machine. In case of WhizNotes, suppose another vendor comes out with a help compiler that uses a document in a format other than RTF, say HTML. It would be easy to change WhizNotes to produce the new format document from your same WhizFolder only because it uses a markup language and is easy to customize.

In fact, many users have asked me for a web authoring tool similar to WhizNotes or [HelpHikes Pro](#). What they like is the multiple-window authoring and hypertext organizational features of WhizNotes and HelpHikes Pro. They are already using WhizNotes for web authoring and using a batch text processor to convert jump statements to HTML links. Very smart! I hope they will like the new WhizMenu feature which will help in HTML coding and customization to some extent. But, I have plans to produce a tool just for web authoring.

4. In order to write a hypertext document in a markup language, all you need is a text editor. You can write your help text for WhizNotes using any editor of your choice and import it into WhizNotes for help generation. This is not possible with WYSIWYG editors which usually work on only one platform for which they are developed.
5. Hypertext authoring tools which use markup languages catch more hypertext-link errors for you up front while doing their own syntax checking. For example, if you use a help authoring tool that uses Microsoft Word as your editor, it can't catch many of the hypertext-link errors. This results in a huge error list given by the help compiler. Not only is it difficult to resolve these errors but the help compiler becomes slower due to the extra exception processing it needs to do. Instead, markup language tools like WhizNotes can be smarter and can even open a topic at the exact position where you made a hypertext related error.

6. It's quicker to evolve or customize the markup languages to take advantages of the new hypertext features. WhizNotes and [HelpHikes Pro](#) even allow you to add your own [HHML](#) statements to the HHML markup language by defining RtfMacros and RtfLinkMacros (available in [HelpHikes Pro](#) only).

For example, in [HelpHikes Pro](#), you can assign HHML statement [**~Legal...**] to mark all the legal text in your document with a style which uses red color with a certain font, a size, a certain logo picture and a help macro that gives a Legal warning to the user. Then later, if your boss doesn't like the color red , it's easy to make a change in the style of all the legal text by just changing the RtfMacro for the HHML statement **Legal**.

Rtf-footnotes

When using WhizNotes, you don't need to understand RTF-footnotes. But, since you're interested, let me give you an idea of how a hypertext jump works internally, in RTF-footnote terms. Suppose you're jumping from a topic "From topic" to another topic "To topic". The top of the "To topic" has been assigned a unique jump-id (**context string**) by using an RTF-footnote marked by character #. Let's assume this **context string** or jump-id is **IDH_TO_TOPIC**. How is the jump statement entered in the "From topic"? It's an underlined text with a hidden text specifying the context string you're jumping to, **IDH_TO_TOPIC**.

Similarly, there are RTF-footnotes for keywords, browsing sequences, topic-titles and so on. It's a real pain in an RTF editor to manage your help attributes as hidden text and footnotes. It's also a big problem to manage these help attributes logically. For example, the help author is responsible for making sure that he assigns a unique **context string** to a jump-target and uses the correct **context string** when entering the jump statement. It can be a real nightmare to maintain hundreds of context strings. Fortunately, with WhizNotes, you don't even have to know the existence of RTF-footnotes or context strings. It handles them by itself, like magic!

Displaying topic title in a non-scrolling region

Please see the top of this topic. You'll see that the title of the topic appears in a non-scrolling region so that even if you page down in the topic, the topic title is clearly visible.

If you want to do that for an individual topic, use the option:

UseNonScrollingTitle = "YES"

in a HelpTopic statement for that topic.

Warning: Note that if you define non-scrolling title for a topic used in a REF statement, the popup won't work properly in WinHelp. The problem is that the Help compiler doesn't give you errors and such problems would pass unnoticed in the generated help. But, WhizNotes is smart and catches all such errors.

Changing the color of the non-scrolling region:

You need to add a window definition for the main window in the [WINDOWS] section of the HPJ file. Here's an example which makes the color of the non-scrolling region as gray. If you want another color, specify its RGB values instead of (192,192,192). Note that some users will be using monochrome screens on notebooks to view your help file, so don't select a color which will look odd on a monochrome screen.

```
[WINDOWS]
main=,,0,,(192,192,192)
```

Since, WhizNotes overwrites HPJ every time, you may want to use various HPJ customization options available in WhizNotes.

All kinds of Jumps & References to another topic

In WhizNotes, you can drag a topic name and drop it on an open topic window to insert a jump to the dragged topic. An Insert Jump Dialog Box appears that allows you to insert a JUMP or a REF statement at that point. When you generate Windows help, a JUMP is converted to a hypertext link and REF is converted into a hypertext popup-link. The format of these statements is:

[~Jump topicname]
[~Ref topicname]

If, instead, you want to copy and paste a jump statement, that method is available too, from an open topic window (menu **Edit-Copy special...**) or from the topic list on a right-click.

Variations of jump links:

You've already seen the simple Jump and Ref statements above. The underlined link-text that is visible in these cases is same as the topic name being jumped to. What if you want to display some other text as the visible link text?

In that case, you should use **[~JumpShowText...]** and **[~RefShowText...]** to show a link-text other than the topic name you're jumping to. The format of these statements is:

[~JumpShowText visible-text::topic-name]
[~RefShowText visible-text::topic-name]

To change a jump to these statements, just type over the changes on a dropped jump statement. Or, you can also use the right-click WhizMenu. For instance, select the visible text, right-click and insert a jumpShowText statement. This will put the caret on the empty topic-name which can be typed in or pasted.

Bitmap hotspots:

Sometimes, instead of showing a link-text, you may want to show a small bitmap. To see how to do that, please see [Bitmaps as hotspots](#).

Jumps to targets within the topic text

This is demonstrated in the sample WZN folder.

Bitmaps as hotspots

If you need to insert a bitmap to represent a single hypertext link, you don't need a hotspot editor. It's more convenient to use the following bitmap statements. We call them 'bitmaps as hotspots'. Also, you can reuse the same bitmap to show it as a hotspot to any number of topics. This is not possible if you had used the hotspot editor.

[~JumpShowBitmap bitmap-file::topic-name]
[~RefShowBitmap bitmap-file::topic-name]

Other variations of these statements are also available for BitmapRight and BitmapLeft:

- **JumpShowBitmapRight**
- **JumpShowBitmapLeft**
- **RefShowBitmapRight**
- **RefShowBitmapLeft**

Also see:

[Bitmaps as hypergraphics](#)

Bitmaps as hypergraphics

There is another amazing tool in your Windows SDK called HotSpot editor **SHED.EXE**. Hotspot Editor allows you to edit your bitmaps to include **multiple hotspot links** in them. When a user clicks on a hotspot link, he can jump to a hypertext topic related to that hotspot. This can be very useful to provide proper help for different parts of a picture. For example, you may show a map where multiple hotspots are defined. Each hotspot pops up a topic which shows information on that part of the map.

For example, here is a hypergraphics picture of a dialog in WhizNotes. Try clicking on any of the items to get help for that item.



Tips on using the HotSpot editor:

Here are the steps on how I created the above hypergraphics using WhizNotes and the HotSpot editor:

1. I added topics "Test dialog", "Test title", "Test page break", "Test Ok" and "Test Cancel" representing each of the items or hotspots on the dialog. Then I entered the text for each of the items into the corresponding topic. Also, I defined HelpTopic statements for these topics with the options **NoTopicTitleAtTop = "Y"** and **TopicNameIsNotAKeyword = "YES"**.
2. I generated an Include file containing Context IDs from Help Generation Dialog Box. This gave me the context strings assigned to each of the topics as IDH_TEST_DIALOG, IDH_TEST_TITLE, IDH_TEST_PAGE_BREAK, IDH_TEST_OK and IDH_TEST_CANCEL respectively.
3. Now I invoked the dialog and captured its image to the clipboard by Alt-PrtSc keys.
4. I started Microsoft Paint, pasted the image and saved it by the file name **EXAMPLE.BMP**.
5. I started SHED.EXE (Hotspot editor) and opened the file **EXAMPLE.BMP**.
6. I dragged the cursor to create a rectangular region over the dialog-image's caption. Then I double-clicked the rectangle to bring up "Attributes" dialog. I entered the context string IDH_TEST_DIALOG, selected type: Pop-up, Attribute: Invisible and pressed OK. This process was repeated for all of the items.
7. I saved the modified bitmap, now a hypergraphics, by the name EXAMPLE.SHG.
8. I inserted a statement [~Bitmap example.shg] in the topic text above.
9. Now I created help from my WZN file and got the results which you can see above.

The HotSpot editor also lets you invoke a Help Macro instead of jumping to a topic.

Note that if your bitmap does not contain multiple hotspots, it's better to use Bitmaps as hotspots.

Changing styles of topic titles

WhizNotes uses its own criteria to display topic titles in a slightly larger font size and a suitable color. If you want to customize the titles to a different format or you want to insert a small picture in front of a title, here is a way to do it.

You can change the RTF for all the topic titles in the [GlobalHelpOptions](#). You will find an example in one of the sample WZN folders.

Customizing Help Project (HPJ) file

WhizNotes generates an HPJ file for compiling your help file. If you're an advanced user and want to edit the HPJ file for any purpose, say for integrating some other RTF files or DLLs with your online help, WhizNotes provides many methods to do that.

Custom HPJ file

You can edit the HPJ file and keep it by a different name to serve as a master copy. Then, you can specify its name in the option `customHpjFile` in the [GlobalHelpOptions](#). Once you do that, WhizNotes will copy this file and use it as the HPJ file instead of generating one.

However, if you use this method, any change on the help generation dialog won't work. For example, if you change the copyright statement, it will have no effect as the HPJ file won't be generated. In this case, you'll have to remember to merge the changes to the master copy. **Hence, it's advised that you don't use this option.** Instead, please use the options described next which allow you to customize the HPJ file sections selectively.

Customizing individual sections of the HPJ file: (only available in the registered version)

To add a new section or to add to an existing section:

If you want to add more lines to a section which is already generated by WhizNotes, then create a topic with the name "`~HpjAddToSection section-name`" where `section-name` is the name of the section to add to. If you do that, any text in that topic will be added to the given section.

Also, if the section is absent, it will be created.

To replace an existing section:

Although, it's not advised, you may want to completely replace a section which is generated by WhizNotes. In that case, put the section text in a topic called "`~HpjReplaceSection section-name`".

Features for context sensitive help

If you are a programmer and want to provide context sensitive help for your program, you will need to know more about the following techniques:

Invoking help topics by WinHelp() function call:

If you're a C or C++ programmer, you can directly code a WinHelp() function call to invoke a help topic. But, you will need to generate the "include file containing Context IDs" when generating help with WhizNotes. It will generate a **.INC** file containing #defines similar to the following:

```
-----  
...  
/* Introduction */  
#define    IDH_INTRODUCTION    259  
...  
-----
```

Here, IDH_INTRODUCTION is the context string or the context id for the topic and 259 is the context number assigned to that topic. The comment above the define shows the topic name for the topic which helps you identify the topic if the context string is not clear or is much shorter than the topic name.

To invoke this help topic "Introduction", you will code a WinHelp call as the following:

```
WinHelp( hWnd, helpFileName, HELP_CONTEXT,  
        IDH_INTRODUCTION);
```

Note that we could have used 259 instead of IDH_INTRODUCTION but that is wrong to do! The whole purpose of generating the **.INC** file is that we can use meaningful identifiers in these calls. Also, if you add or delete topics from your WZN file then a different number will be assigned to the IDH_INTRODUCTION and if you use the number, you will have to change it again and again. Using the context string identifier ensures that you just have to recompile your program if you make change to the WZN file.

Specifying context IDs in application generators:

You might be using an application generator which allows you to assign help topic identifiers to various dialogs and dialog items in your application. In such cases also, you should use the IDH_... identifiers from the **.INC** file. If you are using a language other than C which does not recognize the #define statements, then you can convert the **.INC** file to a format suitable for that language using the **CONVINC.EXE** DOS utility. Examples for VB and PASCAL are given in [Hints on using 'Convinc.Exe' utility with VB and Pascal.](#)

Integrating RTF and Context-id files created by application generators:

Once you know the above technique of using a **.INC** file, you would like to directly code your help topics in the WhizFolder for providing context sensitive help for your application. But if you want to use the **.RTF** file generated by your application generator then you will need to do it in the following steps:

1. You will have to additionally include .RTF file created by the application generator into the [FILES] section of the HPJ file before compiling help. To do that, create a topic with name "~HpjAddToSection FILES" and put the full file name of the other .RTF file in that topic. You can add more file names, each on a different line. For details, please see [Customizing Help Project \(HPJ\) file](#).
2. You will also need to additionally include the context-number file created by the application generator in the [MAP] section of the HPJ file. To do that, create a topic with name "~HpjAddToSection MAP" and put an include statement in that topic for the .H file created by the application generator:

```
#include <C:\APPGEN\APPGEN.H>
```

You can add more such statements, each on a different line. For details, please see [Customizing Help Project \(HPJ\) file](#).

3. Once you do that and recompile your help, it should work except for the following anomaly:

Application generators use their own scheme to assign context strings to the dialog item help topics based on the resource IDs of the dialogs or their items. Very often, these context strings are number-based which is not a good scheme. Application generators may also generate .H files and assign context numbers to the context strings, often the same number as the string itself. In such a case, how do you ensure that WhizNotes generated context numbers don't overlap the application generator's context numbers?

Of course, you can edit either of the header files to change the context numbers. But, you will have to recompile the help with the changed header files. Also, it is very tedious to change these numbers manually.

Fortunately, there is a better solution in WhizNotes scheme. You can change the range of the context numbers generated by WhizNotes by specifying **StartingContextNumber** in the [GlobalHelpOptions](#). It's best to use a number much higher than the highest context number generated by the application generators.

Overriding automatically assigned context numbers:

In some situations, you may need to force the context string or numbers assigned to the help topics by WhizNotes. Here are a few examples of these situations:

- For F1 menu help, you may use an algorithm to calculate a context number for that menu's help topic based on the menu-id. Hence, you want to force the assignment of the calculated context numbers to these topics.
- Similar situation may arise in an application-generator generated code where the context numbers are already assigned based on some scheme and you want to force these context numbers to be assigned to the relevant help topics that you wrote for them yourself in the WhizFolder.
- You have already distributed one version of help with your application. Now you need to distribute a modified help file without redistributing the applications EXE file. If you added or deleted topics such that it changed certain context numbers of the topics invoked by the program, then you need to force the assignment of the context numbers for these topics equal to the older values otherwise the WinHelp() calls from the EXE

won't work.

Of course, this situation can be avoided if you keep the topics list in the non-sorted sequence and always take care to add any new topics to the end of the list. If you follow this procedure then context numbers for the earlier topics will remain same even in the modified help.

In all these situations, you can assign a context number to a topic by using the **ContextNumber** option in the HelpTopic statement for that topic.

Overriding automatically assigned context strings:

Similarly, if needed, you can assign a context string of your choice to a topic by using the **ContextString** option in the HelpTopic statement for that topic.

Avoiding excessive recompilation of the program:

There may be hundreds of help topics in your help file. Whereas, your program may be using context-ids of only ten topics for directly invoking them from the program. Since, WhizNotes generates the **.INC** file containing IDs for all the topics, this file changes as often as you generate your help file. If you directly include the **.INC** file in your program then the Make or Project feature of your program compiler will compile it unnecessarily even if none of the context-numbers for the ten topics used by the program have changed.

The trick is to include a copy of the **.INC** file in your program instead of the original **.INC** file. This will avoid a recompilation of your program. But, remember to compare the copy and the generated file from time to time to see if the context numbers for one of those topics used by the program have changed. If so, synchronize the files by making a fresh copy which will recompile your program.

Of course, the need to synchronize can be minimized if you always keep the topic list in the non-sorted sequence and add any new topics to the end of the list.

How to include #ifndef type statements in .INC file:

You may want to put such statements in the **.INC** file to avoid program compiler errors related to multiple inclusion of this file. But you can't do that as help compiler doesn't like these statements. The trick is to use another **.H** file and put these statements in that file around the **#include** statement for the **.INC** file.

Less often used INI flags

It's possible to change certain behavior of WhizNotes operations which you're likely to change very rarely. This can be done by making a new section **[RareUsageFlags]** in the file **WHIZNOTE.INI** and putting one or more of the following flags in it. Use a value **Y** for On and **N** for Off. By default, all the flags are off.

Note that you will find the above INI file in the Windows directory.

Example:

```
-----  
[RareUsageFlags]
```

```
noLimitOpenWins=N  
-----
```

Flags available:

- **NoLimitOpenWins=Y**

Use it to remove the limit of maximum 50 open windows in WhizNotes. But in that case, if you open a very large number of Windows, memory shortage may result and the behavior of the program is unpredictable in such a case.

- **NoSwitchEditRightClick=Y**

You must have seen the WhizMenu that appears when you right click on an open topic window. On Windows 95, there is a default Edit menu given by Windows 95 that appears on editors by default. In WhizNotes, the default menu appears if you press the **Shift** key with the right-click. It's anticipated that you'll use WhizMenu more, hence this change was forced. If you don't like this, you can switch the menus with this flag so that the Shift key will be used for WhizMenu and the default menu will appear on only a right-click without the shift key.

- **PrintJumpsVerbatim=Y**

By default, WhizNotes prints the jump statements such that only the topic names appear in double-quotes. But if you turn this flag on, Jump statements appear verbatim in the printout.

- **RunHpjCompile=Y**

For DOS based help compilers, by default, WhizNotes makes a BAT file and runs that for help compilation. If you turn on this flag, it still makes the BAT file but doesn't use it. Instead, it tells Windows to perform an open operation on the **HPJ** file using the configured program. This flag is automatically turned on by WhizNotes if it finds out that you are using the new Windows based HCW compiler.

- **HpjNoOemConvert=Y**

For the DOS based help compilers, WhizNotes converts your HPJ options, such as the copyright statement, to the installed OEM character set so that no problems occur for foreign language users. If this flag is turned on then no OEM conversion takes place. This flag is automatically turned on by WhizNotes if it finds that you are using the new Windows based HCW compiler.

- **NoResolveContextClashes=Y**

For similar topic names, it may happen that the generated context strings are same. By default, WhizNotes resolves such a clash by appending a number suffix to the context string. If you turn this flag on, WhizNotes will show you the clash error and won't proceed. The only solution will be to change one of the topic names and proceed. The advantage of turning this flag on is that the context strings will be definitely derivable from the topic names across multiple help files generated by WhizNotes. Future versions of WhizNotes will use this feature to make it possible to process JumpWzn statements for hypertext jumps across help files created from different WZN files.

- **TestHelpFromShell=Y**

If you're using the new HCW compiler on Windows NT 3.51 to generate WinHelp 4.0, you may see that when WhizNotes invokes your generated help for testing, it is invoked with old 3.1 WinHelp. This problem occurs because of the way WinHelp 3.1 is set up (bug?) in Windows NT 3.51 environment. If you turn the above flag on then correct WinHelp will be invoked.

- **DontGrabWZNExtension=Y**

This flag is available with all of the author's programs which can process WhizFolder documents. Usually, on startup a program grabs the WZN extension if it has been assigned to another program so that subsequently, when you launch a WZN document from the desktop shortcut or from the Explorer or File Manager, the program used last is started up for opening the WZN document. This is often the most desirable behavior. However, if you are using more than one of author's programs and always want only one program to start from a desktop WZN shortcut, you should turn on this flag in all **other** programs.

- **UseWorkTempDir=Y**

See [Please check the program's speed.](#)

- **InsertTimeOnly=Y**

If you put on this flag, the Edit menu operation "Insert Date/Time" starts working as "Insert Time" only. You can still get both by pressing F4 and F5 in succession.

Modifying RTF and HPJ files with another tool

WhizNotes preserves the RTF and HPJ files after the help compilation. You can use any other tool to change these files and rerun the help compilation to get the changes in your help.

The RTF file:

You may want to change the RTF file if you want some text formatting feature not supported by HHML statements. However, please remember that next time you generate help from WhizNotes, it'll overwrite the RTF file. To avoid that, it's best to use another file name prefix for the RTF and HPJ file for making the changes.

If you know RTF, you can also make new HHML statements for the desired formatting using the RtfMacro feature of WhizNotes. For an example, please see the sample WZN files where certain RtfMacros are defined for making a tabular display.

The HPJ file:

You may want to change the HPJ file if you're using the new HCW compiler to customize the HPJ file or you want to use 3rd party DLLs in your help. In that case, to preserve your changes, either use a different HPJ file name or select the option in WhizNotes not to overwrite the HPJ file. However, it's advised that you merge the changes made by HCW into the HPJ file generated by WhizNotes by using the HPJ customization features.

Setting up print margins and other printing options

This dialog appears when you select the menu "Options-Printing options...".

Here's a description of the items available on this dialog:

Print margins:

It allows you to set up various margins for the printout, in inches.

Spacing of lines:

It allows you to alter the spacing of lines where two successive printed lines are spaced by the number of lines indicated. The default value is 1.

Controlling printout of multiple-topics:

It allows you to control how multiple topics are printed. You can control the formatting of the topic names in a multiple-topic printout.

You can also control how many minimum beginning lines of a topic must be present on the same page with the topic name. This is done by entering a value for the option "Keep beginning lines on same page". The value is used to determine whether so many beginning lines of the next topic will fit on this page. If they don't fit, the topic is printed on the next page so that these lines are together with the topic name.

You can also control whether the header, footer, table of contents and the date/time on the table of contents is printed or not printed.

Also see:

[How jumps are printed](#)

The magic of Hypertext links

There are various kinds of hypertext links in Windows help and there are numerous ways to define and control them in WhizNotes. Here are the topics that you should read for more details on various types of hypertext links available in WhizNotes.

[All kinds of Jumps & References to another topic](#)
[How to insert bitmaps into your help files](#)
[Defining keywords for topics](#)
[Browsing sequences of topics](#)
[Inserting help macros](#)
[Specifying jumps & references to other help files](#)

The power of the HelpTopic paradigm

WhizNotes gives you a logical grouping mechanism by which you can keep all the keywords or macros of the related topics close to each other. This is done in an object oriented manner by treating a topic as a **HelpTopic** object and treating the keywords and macros as attributes of that object.

You simply code a HelpTopic statement for a topic and assign the **keywords, macros** and other attributes to it. Even if you don't assign, there are convenient defaults. For example, WhizNotes always defines one keyword for the topic which is same as the topic name. You keep the HelpTopic statements for a group of topics together in a separate topic with a special name starting with "~HelpAttributes". You can also quickly locate a HelpTopic statement for a topic by right clicking on the topic list.

A HelpTopic statement also lets you assign other useful options of a topic, all in one place and in free-format text. For details, see Keywords, macros, and other characteristics of topics.

WhizMenu for quick entry of canned text and HHML statements

Introduction to WhizMenu

WhizNotes contains a very smart feature to quickly paste Canned text in your topic. For an example, just right-click on an editor window and you will see a menu appear on the right. Click on an item and the item pastes its text in the editor. This is a customizable menu, and you need to add your own items to it. What's more, you can have several different WhizMenus and switch between them.

There can be several uses of it. If you often require the use of some text, such as a letter or a form, keep it in WhizMenu so that you can quickly paste it. Another nice feature allows you to enclose the selected text at a desired place in the Canned text. For example, this can be useful to create HTML statements for your own use.

By default, WhizNotes uses the WhizMenu file WZMENUR.WZN in its program directory. This WhizMenu is for help generation users and allows quick entry of HHML statements. If you are not a help generation user, you should switch to the other WhizMenu file WZMEXAMP.WZN. To do that, right-click on an editor window to bring up the WhizMenu, then click on "WhizMenu--Switch to WhizMenu file" and select WZMEXAMP.WZN in the file open dialog box that comes up. When you click OK, that file controls your right-click menu.

WhizMenu for general use:

To put your own Canned text, right-click on an editor window, and select "WhizMenu..." submenu, then click on "edit the current WhizMenu". This will open a WhizFolder which controls this menu. You can also switch to another WhizFolder to act as your WhizMenu.

If you are a help authoring user, you can skip this section and read the next one WhizMenu for HHML statements. If not, first switch to the WZMEXAMP.WZN whizmenu as given above. Then, to get started, it's best to use the examples in the menu items to understand these features. To see it, just Right-Click on an empty editor window and click on the items one by one and read them.

WhizMenu for HHML statements:

WhizNotes contains a very smart feature to quickly enclose the selected text in an HHML statement. Just select some text in an open topic window, right-click or press Alt-M to bring up a special menu and select the statement you want.

What's more, you can customize this menu to change the current statements as well as to add new statements that you might have added with RtfMacros. To add or change the items, right-click and select "WhizMenu..." submenu, then click on "edit the current WhizMenu". This will open a WhizFolder which controls this menu. You can also switch to another WhizFolder to act as your WhizMenu. So, you can keep different WhizFolders for different purposes.

You may also want to look at the other WhizMenu for general users which also gives examples of HTML statements for web authoring. Please read the top of this topic giving notes for general users.

Steps to prepare a new WhizMenu folder:

WhizMenu folders are prepared in these steps:

- The default WhizMenu folder is WZMENU.WZN in the WhizNotes program directory. To make your own folder, copy this file by another name starting with WZM in the same directory. Then use WhizMenu submenu to switch to the other WhizMenu file. Open and see a few of the topics to see how they are formatted. A description follows.
- The individual statements use free-format text to describe their entry. In the text, %SELTEXT% is used to mark the position where the selected-text will be inserted. You can additionally use a %CARET% to mark the place where the caret will appear after the entry. By default, the caret appears at the %SELTEXT% position.

For an example, see the topics for 'Bold' and 'Startlist'. Note that the text defined for a statement topic is stripped of the beginning and ending newline characters and spaces. But if you want to include the ending or starting new lines (see Startlist), put a %SENTRY % at the beginning or end to prevent that. The sentry guards the new lines from stripping.

- When adding your own statements, use an ampersand before the character which you want to work as the shortcut-key for the menu. Remember, it must be unique. If you can't come up with a unique key, leave it without ampersand. In that case, you will have to select it with a mouse or by using the arrow keys.
- You can cause menu separators (bars) to appear in the menu by inserting dummy topic names that begin with %SEP%.
- You can enter submenus by inserting them between the topics "%STARTPOPUP% popup-name" and "%ENDPOPUP%".

Using WhizNotes as a hypertext organizer

Manipulating WhizFolder files:

- [What is a topic?](#)
- [What is a WhizFolder file?](#)
- [Creating a new WhizFolder file](#)
- [Opening an existing WhizFolder file](#)
- [Getting WhizFolder details by right-click](#)
- [Using 'Separator' topics to organize the topic list](#)
- [Changing the font size for the topic list](#)
- [Opening from File Manager](#)
- [Creating shortcuts to open WZN files](#)
- [Saving a WhizFolder file](#)
- [Desktop options for a WhizFolder](#)
- [Save WhizFolder as.. menu option](#)
- [Edit menu copy operations for a WhizFolder window](#)
- [Obtaining word count](#)

Manipulating topics within a WhizFolder file:

- [Creating a new topic](#)
- [Opening a topic for editing](#)
- [Features of the topic editor](#)
- [Renaming a topic](#)
- [Deleting a topic](#)
- [Printing from a topic window](#)
- [Printing one or more selected topics](#)
- [Setting up print margins and other printing options](#)
- [Ordering your topics in the topic list](#)
- [Searching for a topic](#)
- [Search and select operation](#)
- [Search/replace operations](#)
- [Inserting a jump to another topic](#)
- [Operations on multiple topics](#)
- [Inserting date/time in your text](#)

Making use of Windowing features of WhizNotes:

- [Opening multiple WhizFolder and topic windows](#)
- [Manipulating external text files](#)
- [Saving topic to an external text file](#)
- [Copying topics from one WhizFolder to another](#)
- [Options menu](#)
- [Toolbar/Statusbar](#)

Printing WhizFolder files and topics:

- [Basics of WhizNotes printing facilities](#)
- [Setting up print margins and other printing options](#)
- [Printing from a topic window](#)
- [Printing one or more selected topics](#)
- [Using Quick draft printing for convenience](#)

Export/import features of WhizNotes:

- [Exporting selected topics to a text file](#)
- [Importing topics from a text file](#)
- [Importing whole text files as topics](#)

Exporting selected topics in RTF format

What is HHML?

WhizNotes uses a special *hypertext markup language* called HHML which is created by the author of WhizNotes.

Here are the basics of HHML:

- An HHML statement looks like [**~statement** enclosed-text]. The examples are:

[**~jump** topic-name]
[**~bold** enclosed-text]

The first is an example of a hypertext link where the enclosed-text specifies the topic-name to jump to. The other case is an example of the **bold** attribute which can be applied to the enclosed-text. The enclosed-text can be as long as a whole para. If the enclosed-text is absent, it works as a flag and makes all the remaining paragraph bold unless it encounters a [**~wResetFormat**] statement.

Tips:

1. It's good to know that all the help statements, attributes, or options used in HHML are case-insensitive; for example, you can use "JUMP", "Jump", or "jump" when specifying jumps.
 2. **Embedding ']' character:** If you want ']' character to be part of your enclosed text then prefix it with a backslash. This applies to all the HHML statements.
- The list of HHML statements supported by WhizNotes can be found in [HHML: List of statements supported by WhizNotes](#).
 - You can define your own HHML statements by using [RtfMacros](#).
 - For easy data entry of HHML statements, built-ins and your own, a right-click WhizMenu is provided which is customizable. See [WhizMenu for quick entry of canned text and HHML statements](#).
 - The [HelpTopic](#), [ListBrowseGroup](#), [GlobalHelpOptions](#), [RtfMacro](#) are not HHML statements. Rather, we call them HHML tools. Because they support various features indirectly related to HHML.

You'll notice that some statements use a 'w' prefix. Why? Please see [Why some HHML statements have 'w' prefix?](#)

Where to get the help compiler

Do you want to create the old or the new version help files?

Which help compiler you use depends on whether you want to create the old or the new (Windows 95) version Windows help files. The old version help files can be used on Windows 3.1 as well as on Windows 95. However, they don't have the new features found in Windows 95 help. On the other hand, the new Windows 95 or version 4 help files have the new features but they can't be used on Windows 3.1 systems.

The old version help files are created with the DOS based help compilers whereas, the new version 4 help files are created with Microsoft Help Workshop (HCW) which is a Windows based compiler with a GUI interface. To summarize, if you are creating help for a 16 bit application which can be used on both Windows 3.1 and Windows 95, it's best to create the old version help files by using a DOS help compiler. If your application targets only Windows 95 or NT platforms, you can create the new version 4 help file using the new HCW compiler.

Now, which compiler WhizNotes uses simply depends on what compiler EXE file is associated with the HPJ extension on your system or what you specify in the GlobalHelpOptions statement to override the above association. Read more about this set up and where to get the help compilers in the following discussion.

DOS based help compilers:

If you already have a Windows SDK or an equivalent, chances are that you already have the help compiler. See your SDK documentation for more details. The file name is likely to be HC31.EXE or HCP.EXE and you're likely to find it in the same directory where your programming language compiler is present. In any case, it's better to get the newer HCP.EXE compiler as given in the following paragraph.

If you use HC31.EXE then you might experience memory shortage when compiling the help files containing large bitmaps. The solution is to get a protected mode version of the help compiler, HCP.EXE, that uses extended memory. On Internet, you can try this location as per the Microsoft Software Library catalog: S14344 HC505.EXE, Latest Version of 3.1 Help Compiler:

<ftp://ftp.microsoft.com/softlib/mslfiles/HC505.EXE>

Hint: Just enter the above URL in your web browser and you will be able to download the above file. The case is important for the file name. So, type exactly as shown.

Installation instructions: HC505.EXE is a self-extracting archive so run it in a temporary directory to unzip. After unzipping, move the files to a suitable directory, preferably on your path. Then associate the HPJ extension in your Explorer or File Manager with the full file name of HCP.EXE. You need to use the full file name if HCP.EXE is not on your DOS path. If you are using Explorer, use the "View--Options" menu to associate the HPJ extension with the HCP.EXE file. Once you do that, WhizNotes will automatically use the compiler when you "Create help" from a folder on the Tools menu.

Windows based help compiler:



The new HCW compiler is a Windows based compiler for creating Windows 95 help. Previously, it used to be available only with the newer Windows SDKs or with a help authoring book by Microsoft. But, now Microsoft has made the latest version available on the

following location:

<ftp://ftp.microsoft.com/softlib/mslfiles/hcwsetup.EXE>

Hint: Just enter the above URL in your web browser and you will be able to download the above file. The case is important for the file name. So, type exactly as shown.

Installation instructions: hcwsetup.EXE is a self-installing archive so just run it and it will let you install the Microsoft Help Workshop (HCW). When you run the HCW, it automatically sets up the association for HPJ with itself in Explorer. So, once you run it, WhizNotes will automatically use the compiler when you "Create help" from a folder on the Tools menu. However, if you are using HCW for some folders and want to use the DOS based compilers for other folders, this automatic HPJ association by HCW will cause problems for you. The solution to deal with this problem and other details on how WhizNotes really uses HCW is given in the topic [Using the new HCW compiler](#).

HHML: List of statements supported by WhizNotes

This topic serves as a reference for all the HHML statements and tools. It's very useful, and you may define a bookmark here so that you can come to this page quickly. You'll notice that some statements use a 'w' prefix. Why? Please see Why some HHML statements have 'w' prefix?

The statements and tools are grouped by functionality in the following list.

- Hypertext links
- Pictures, Hypergraphics
- Text attributes/Styles
- Lists and indentation
- Margins/spacing/paragraphs/comments
- Conditional compilation
- HHML Tools
- For advanced RTF users
- Special topic names

Hypertext links:

- ~Jump - A jump where the visible link is same as the target topic name.
- ~Ref - A popup reference where the visible link is same as the target topic name.
- ~JumpShowText - The visible link is different from the target topic name.
- ~RefShowText - The visible link is different from the target topic name.
- ~JumpShowBitmap - The visible link is a bitmap.
- ~RefShowBitmap - The visible link is a bitmap.
- ~JumpShowBitmapLeft - The visible link is a left flushed bitmap.
- ~RefShowBitmapLeft - The visible link is a left flushed bitmap.
- ~JumpShowBitmapRight - The visible link is a right flushed bitmap.
- ~RefShowBitmapRight - The visible link is a right flushed bitmap.
- ~JumpToContext - Jump to another help file or to the middle of topic.
- ~RefToContext - Ref to another help file.
- ~JumpWzn - Provides a hypertext organizer feature to jump to another WhizFolder's topic. However, this is not a help generation statement and can't be processed by the help generation logic. In future, it might provide the capability to jump to a Help file generated from the other WhizFolder.
- ~HelpMacro - The link executes your help macros defined here.
- ~wKeywordsHere - Lets you insert keywords in the topic text.
- ListBrowseGroup - A tool to define browse groups and sequences of topics.
- HelpTopic - A tool to assign various attributes or options to a topic object.

Pictures, Hypergraphics:

- ~Bitmap - To display a bitmap.
- ~BitmapLeft - To display a bitmap flushed left in the paragraph.
- ~BitmapRight - To display a bitmap flushed right in the paragraph.
- Also see Bitmaps as hotspots and Bitmaps as hypergraphics.

Text attributes/Styles:

- ~Bold - Bold text.
- ~Italic - Italic text.
- ~wBoldItalic - Bold and italic text.

- ~wSmallCaps - Text in small capital characters.
- ~wStrikeout - Striked-out text.
- ~wUnderline - Underlined text.
- ~wResetFormat - Restore the default character properties when above statements are used as a flag without enclosed text.

Lists and indentation:

- ~Indent - Indent the paragraph.
- ~StartList - Start a bulleted, Numeric, or NumericCompact list.
- ~Item - Start a list item.
- ~EndList - End the list.

Margins/spacing/paragraphs/comments:

- ~wNextLine - Start a new line without changing the paragraph.
- ~wComment - To insert comments in the help text which are ignored when processing.

Conditional compilation:

- ~IfTag - Include the following text if this tag is defined in the Help Generation Dialog Box.
- ~IfNotTag - Include the following text if this tag is NOT defined in the Help Generation Dialog Box.
- ~Else - Else text for the above statements when they are not true.
- ~Endif - End of the text for conditional include.

HHML Tools:

An HHML tool is a statement which can't be directly used in your topic text. Rather, it's usually defined in a specially named topic and indirectly controls some aspect of the help generation. An example of an HHML tool is a **HelpTopic** statement.

- HelpTopic - A tool to assign certain attributes or options to the topic objects.
- ListBrowseGroup - A tool to assign the browsing groups and sequences to a set of topics.
- RtfMacro - A tool to create custom HHML statements.

For advanced RTF users:

- ~RTF - Insert direct RTF code in the help text.
- ~RtfOn & ~RtfOff - Insert a multiple-line block of RTF code in the help text.
- RtfMacro - A tool to create custom HHML statements.
- Also see some RTF options in the GlobalHelpOptions which allow you to customize the RTF for topic titles.

Special topic names:

- "~~..." - Comment or separator topics. They are ignored in help generation.
- "~HelpAttributes..." - Used to define the HelpTopic and ListBrowseGroup tools.
- "~RtfMacros..." - Used to define the custom HHML statements.
- "~StartupHelpMacros" - Used to define the Help Macros which execute when the help file is first displayed.
- "~HpjAddToSection section-name" - Either add a new section in the HPJ file or add to an existing section.
- "~HpjReplaceSection section-name" - Completely replace a section in the HPJ file.

Starting a new line within a paragraph

The statement [**~wNextLine**] starts a new line within a paragraph.

Why would you like to do that? I can give you an example:

Suppose you insert a bitmap to be displayed left flushed in a paragraph. The text will then appear wrapped to the right. What if your text contains only the jump statements and you want these jump statements to appear each on a new line, to the right side of the bitmap? You'll have to prefix each jump with the above statement.

Using the new HCW compiler

You need not read and understand the following discussion unless you are really interested in controlling some aspects of HCW usage. If you want to use HCW for some of your WhizFolder files and DOS help compilers for other WhizFolder files, you should read the section "How to change from HCW to a DOS based compiler?" described at the end of this topic.

How WhizNotes uses HCW:

Every time you run HCW, if needed, it installs itself by registering an entry "Help Project File" in the Windows 95 registration database. WhizNotes automatically detects that and uses HCW. **But since HCW is a Windows based program, WhizNotes performs certain operations in a different way:**

No OEM translation to HPJ file:

WhizNotes doesn't translate ANSI characters to OEM in the HPJ file as the Windows based HCW compiler can process ANSI characters. This is relevant only if you're using special characters or foreign characters in your title or copyright strings. Foreign language users should specifically note this, and they should not attempt to use this HPJ file on the DOS prompt with a DOS help compiler. The characters in HPJ file won't be processed correctly. Whenever you switch back and forth between a Windows and a DOS help compiler, it's best to always use WhizNotes to regenerate the HPJ file.

No wait for compilation:

WhizNotes starts the compilation but can't know when it finishes because the HCW compiler is a Windows program and keeps running even when the compilation is over. Hence, WhizNotes doesn't wait for the compilation to finish. WhizNotes uses the command line switch '/C' to start the compilation. By default, it also uses the switch '/R' so that as soon as the compilation finishes, the help file is invoked.

Tip for advanced users: You can specify your own command line switch combinations in the INI parameter **HcwCmdLineSwitches**, which can be added in the [INIT] section of the **WHIZNOTE.INI** file. Note that if you don't specify the '/C' switch, it's forced.

You will find the WHIZNOTE.INI file in your Windows directory. Here's an example of coding the switches in the INI file:

```
[INIT]
...
...
HcwCmdLineSwitches=/C /R
```

How to change from HCW to a DOS based compiler?

Technically, it's possible to change to a DOS based compiler, say HCP.EXE by changing the HPJ association in the Explorer (menu View-Options...). WhizNotes, by default, always uses the compiler associated to the HPJ extension. **But there is a problem.** If you ever start HCW, it will again associate itself with the HPJ extension and you will have a hard time changing the explorer options again and again.

Solution: To solve the above problem, WhizNotes gives an easier way to specify a compiler for a WZN file. Let HCW be your default help compiler. If you ever want to use another

compiler for a particular WZN file, please specify the compiler's file name in the parameter **ForceHelpCompilerToUse** in the GlobalHelpOptions.

Example:

```
GlobalHelpOptions
{
    forceHelpCompilerToUse = "HC31.EXE"
}
```

This way, you can use different compilers with different WZN files without changing the association in the Explorer. Also, for large WZN files you can always use the option No wait compilation and start all the compilations simultaneously in different windows, even using the different help compilers.

Displaying text in tables

For simple text, you might get away with using Tabs in your topic text. However, in your WhizFolder, if you are using a font other than "MS Sans Serif", size 10 or are using a tabsize which is not 6, you should select the option "Font/tabsize - Same as WhizFolder" on the help generation dialog. Otherwise, your tabs as displayed in the topic window won't match those displayed in the generated help.

Better use Table RTF:

If the above method doesn't work for your table data, it's best to use the RTF features available for tables. WhizNotes does not give direct HHML statements for creating tables because there are too many variations possible and it's difficult for WhizNotes to diagnose and catch all the table related errors in your topic text. Instead, it's recommended that you use the RtfMacros feature to define your own HHML statements and define the tables in any way you want.

For an example of using the tables, please see the topic "Creating tables" in the sample WZN file.

Creating shortcuts to open WZN files

Shortcuts are very convenient to open WZN files. They'll open a WZN file for you even if WhizNotes is already running.

On Windows 95, you can create a shortcut for a WZN file on the desktop. The quickest way to do that is to drag a WZN file from the Explorer to the desktop with the mouse right button.

On Windows 3.1, you can do that by dragging a WZN file from the File Manager on to a Program Manager group window.

Also see:

[Startup Tips and Tricks](#)

Getting WhizFolder details by right-click

To get information on a WhizFolder window, you can right-click on an area outside the topic list. It will show you useful information on the total, visible and selected topics and will also show the full name of the WhizFolder file.

If you right-click on the topic list then a menu appears that allows you to do various operations on the selected topics. For example, you can quickly locate the HelpTopic statement if you have selected only one topic. Or, you can copy [~Jump ...] statements for the selected topic names to the clipboard.

Search and select operation

If you select the menu **Search-Select topics containing text**, you can select a set of topics which contain the search text in the topic name or in the topic contents.

Convert into paragraph

Normally, you'll need a word-wrapped display whenever entering or editing normal text in your topics. This is mandatory if you are creating topics to be converted to online help or for use in a word processor.

However, at times you will require to import or paste text from the external text files. The paragraphs in these text files may not be true paragraphs, especially if they are prepared with DOS editors or programmer's editors. In other words, the paragraphs will be broken into multiple lines by hard carriage returns at the end of each line.

Convert into paragraph operation (Edit menu) converts the selected text to a word-wrapped paragraph by removing the hard carriage returns. You can even select the whole text of the window and perform this operation. Don't worry, it won't create a single big paragraph in this case. Rather, it works by its own logic, trying to recognize where a paragraph might end and a new paragraph might start. You can even undo the operation if you don't like the results.

Advanced usage: preserving tabular text:

If this operation encounters embedded tabs or an extra space characters, it gives you a warning before stripping them so as not to disturb your formatting. Hence, if you have tabular text or the text where intervening tabs or blanks are to be preserved, you will need to be careful.

The best way to preserve tabular text is to do the conversion by selecting pieces of text which you actually want to convert to paragraphs. This way, you can avoid selecting the tabular text.

But, suppose, you have a large body of text, or you often generate text from some other tool to be imported into WhizNotes. In such a case, you can use special keep indicators to prevent pre-formatted text from conversion. You should enclose such text in the keep indicators before performing this operation. A keep indicator is a line beginning with "-~" characters. Initially, keep is off but each successive keep indicator toggles the status of the keep. You can actually keep track of the keep being On or Off by noting it to the right side of the keep indicator. After the conversion, you can remove all the keep indicators. Note that you are likely to use the keep indicators only if you do this very often and use a tool to generate the text for import. Otherwise, you are better off doing the conversion selectively.

Changing the font size for the topic list

You can enlarge or reduce the font for the topic list by the menu **Edit- Enlarge/Reduce topic list font**.

You can also put up a horizontal scroll bar on the topic list by the menu **Edit-Enable horizontal scrolling**.

Features of the topic editor

When you open a topic for editing, the regular features of a Windows editor are available. In addition, the following features are available:

- You can insert hypertext links to jump to other topics or to the topics in other WZN files.
- You can use the menu **Edit-Copy special...** to copy the topic name to the clipboard. You can also copy a JUMP statement or a REF statement for the topic to the clipboard so that you can paste it in another topic as a link.
- When you right-click, a WhizMenu appears that allows you to conveniently enter Canned text or HHML statements in the topic text.
- Auto indent is available on the Edit menu.
- Word wrap is available on the Edit menu.
- You can press **Ctrl-Y** to delete a line.
- You can double-click at the beginning of a line to select the whole line.
- You can insert the date/time in long or short format from the Edit menu.
- You can count the total words from the edit menu.
- You can select a block of text and convert it into a paragraph.
- You can save the topic to a text file such that the paragraphs are broken into lines as shown in the window.

Defining keywords for topics

When you press the "Search" or "Index" button in Windows help, you actually use a keyword search facility where a dialog box appears showing you a list of phrases or words. These are called keywords. If you're using Windows 95, the keywords appear on **Index** tab of the **Help Topics** dialog.

Note that you can define multiple keywords for a topic so that by selecting any of these keywords in the keyword index, the user can jump to that topic. Also, the same keyword can be defined for a number of topics. In that case, on selecting the keyword, the user gets a list of the topics to choose from.

WhizNotes gives you two ways to define keywords.

Defining keywords by a HelpTopic statement:

You can define one or more keywords in the HelpTopic statement for a topic. *Make sure that if you are defining multiple keywords for a topic, you must separate them with a semicolon.*

This way, it's easier to compare the keywords for different related topics as the HelpTopic statements are free-format and you can define them together in groups of related topics. Also, by default, WhizNotes assigns a keyword to each topic which is same as the topic name. If you don't want that to happen for a particular topic, you can put off the option **TopicNameIsNotAKeyword** in the HelpTopic statement for that topic. For more details, please see Keywords, macros, and other characteristics of topics.

Defining keywords directly in the topic text: (only available in the registered version)

If you define the keywords by the previous method, then selecting the keyword in the keyword index takes you to the top of the topic. You can also define a keyword in the text of the topic so that the index takes you to that exact place in the topic. This is done by using the HTML statement **wKeywordsHere** directly in the text of the topic. The format of this statement is:

[~wKeyWordsHere keyword 1;keyword 2;...]

Make sure that if you are defining multiple keywords for a topic, you must separate them with a semicolon.

This dialog appears when you print more than one topic from a WhizFolder.

This title is printed on the header of each page and on the "Table of contents".

By default, topics are printed together if they can be accommodated on a page. But you can check this option if you want each topic to be printed on a new page.

Press OK to proceed with printing.

Press Cancel to discontinue this operation.

An HHML tool is a statement which can't be directly used in your topic text. Rather, it's usually defined in a specially named topic and indirectly controls some aspect of the help generation. An example of an HHML tool is a **HelpTopic** statement.

For a complete list of tools, please see [HHML: List of statements](#).

How jumps are printed

By default, WhizNotes prints the Jump statements such that only the topic name appears in double-quotes. But you can change that by an INI flag if you want the Jump statements to print verbatim. For details, please see [Less often used INI flags](#).

Print list of topics

This operation on the menu **File-Print list of topics** prints a list of all the topics in the WhizFolder.

Copy Jump statement

Copies a Jump statement for the selected topic to the clipboard so that you can paste it in another topic at a suitable place.

If you have selected multiple topics, multiple Jump statements are copied.

This action is also available on the Right-click menu on the topic list and from an open topic window's Edit menu.

Also see:

[Getting WhizFolder details by right-click](#)

Copy Jump statement for this topic

Copies a Jump statement for this open topic to the clipboard so that you can paste it in another topic at a suitable place.

This action is also available on the Right-click menu on the topic list and on the WhizFolder window's Edit menu.

Also see:

[Getting WhizFolder details by right-click](#)

Copy Ref statement

Copies a Ref statement for the selected topic to the clipboard so that you can paste it in another topic at a suitable place.

If you have selected multiple topics, multiple Ref statements are copied.

This action is also available on the Right-click menu on the topic list and from an open topic window's Edit menu.

Also see:

[Getting WhizFolder details by right-click](#)

Copy Ref statement for this topic

Copies a Ref statement for this open topic to the clipboard so that you can paste it in another topic at a suitable place.

This action is also available on the Right-click menu on the topic list and on the WhizFolder window's Edit menu.

Also see:

[Getting WhizFolder details by right-click](#)

Enlarge topic list font

Enlarges the font size for the list of topics. The size can not be enlarged beyond a pre-determined highest size.

Reduce topic list font

Reduces the font size for the list of topics. The size can not be reduced beyond a pre-determined lowest size.

Enable horizontal scrolling

If you check this menu option, a horizontal scroll bar appears on the list of topics to enable complete viewing of long topic names.

Show list status

Shows useful information on the total, visible and selected topics and also shows the full name of the WhizFolder file.

You can also get these details by right-clicking on the WhizFolder window outside the topic list. If you right-click on the topic list itself, this action is also available on the popup menu that appears.

Also see:

[Getting WhizFolder details by right-click](#)

Locate HelpTopic definition for this topic

Quickly locates the HelpTopic statement for this topic so that you can modify it. However, while looking for the desired topic, if it finds errors in another topic's HelpTopic statement, it gives an error message and can't proceed till you correct the error. This is done so that a spelling-mistake in HelpTopic name even for the current topic can be detected.

This action is also available on the Right-click menu on the topic list.

Also see:

[Getting WhizFolder details by right-click](#)

Copy topic name

Copies the topic name to the clipboard so that you can paste it somewhere else.

This action is also available on the Right-click menu on the topic list and on the WhizFolder window's Edit menu.

Also see:

[Getting WhizFolder details by right-click](#)

Why some HHML statements have 'w' prefix?

You must have noticed that some HHML statements have a 'w' prefix whereas some don't. For example, ~**wBoldItalic** uses a 'w' prefix whereas ~**Bold** does not use a prefix.

This is a legacy of earlier versions of WhizNotes from which HelpHikes has evolved. Initially, a limited HHML set of statements was given in WhizNotes. But users were also given RtfMacros to define more statements of their own. An example for **BoldItalic** was provided as a user defined statement.

Now, HelpHikes has to support all the previous WZN files created for WhizNotes help generation. So, when HelpHikes HHML language was extended, I didn't want the new HHML statements to clash with the existing RtfMacros devised by the users. Hence, 'w' prefix was added to most of the new statements.

Hints on testing the help file

If you are using the new HCW compiler, it has got many features for testing various aspects of the compiled help file. For details, please see the HCW documentation.

Visiting all the topics:

However, at times, when testing your HLP file, you may wonder whether you have really visited all of your topics to check them. With many of the topics buried several levels deep in your hypertext document, this is not an easy task. Here's a trick which I use when checking all of my topics in the compiled HLP file:

1. For this trick to work, you need to define the browse sequences in a separate topic. If you haven't done so, create a new topic "~HelpAttributes for browse groups" and move your ListBrowseGroup statements to that topic.
2. Now, temporarily rename the above topic to make it a comment or separator topic. For instance, add an additional '~' character to prefix the topic name. Now generate help. This will cause all the browse groups to be ignored. Also make sure that on the help generation dialog, you have not checked the option "Remove browse buttons".
3. Also comment out your Contents topic if any by prefixing it with ~~.
4. The nice thing is that when WhizNotes finds no Browse Groups present, it includes all the topics in a big browse group where the topics appear in the same order as in your WhizFolder file, except that the Contents topic is always made the first topic.
5. Now your job is much easier. Just start the online help and go to the first topic Contents.
6. After verifying the Contents topic, just press the '>>' browse button to move forward and visit all of your help topics one by one. They'll appear in the same sequence as your WhizFolder, and if you find a problem, you can easily find that topic in your WhizFolder and correct it before proceeding.
7. If you interrupt your work for some reason, just note down the last topic name checked. Then, on next start up, you can easily find the last topic in the Contents created by WhizNotes which is in the same order as the natural browsing sequence. Open that topic and resume from there.
8. Once your testing and corrections are over, rename the above browse-group topic and the Contents topic back to the original name so that it's not ignored in the next compilation. Be sure to recreate the help so that the browsing orders are restored.

Note that this method only allows you to check the text contents of all the topics. It also lets you verify the visual aspects of the topics, unless you are using secondary windows. To check the visual aspects of the topics which use a window other than "main", you'll have to visit them from a jump statement so that they are displayed in the proper secondary window.

Convert to lower case

Converts all the letters in the selected text into lower case.

Convert to upper case

Converts all the letters in the selected text into upper case.

Check spelling

(only available in the registered version)

When you click on the menu **Tools-Check Spelling**, spelling check is performed on the text in the current window or in the current WhizFolder. You can change options for the Spelling Checker from the options menu or from the Spelling Checker dialog. If you have installed more than one language dictionaries, you need to run Setup to change to another dictionary.

Spelling-checker options...

Allows you to change the Spelling-Checker Options.

Force 'ignore all', 'change all' each time

This menu item is located on the Tools menu and can be checked or unchecked by clicking on it. If unchecked, it remembers your 'ignore all' and 'change all' preferences across different Spelling-Check sessions. If checked, it forgets them in the next session.

Note that when you perform a help generation, the Spelling Checker is unloaded to save on memory. Hence, all the 'ignore all' and 'change all' preferences are forgotten.

What you get when you Register

Please don't be deceived by the simple user interface of WhizNotes. *Complex* objects are at work, behind the scenes, to make it much easier for you. Please order the registered version and support this innovative development. You get a choice of ordering the registered version of one of the three similar programs, depending on your needs. These programs are WhizFolders, WhizNotes and HelpHikes Pro. Now that you have seen what WhizNotes can do for you, you should also see what is different in WhizFolders or HelpHikes Pro.

Goodies for registered users:

Here is what you get with the registered versions of the programs. Note that the help generation features described here are not available in WhizFolders. They are only available in WhizNotes or HelpHikes Pro as indicated.

- **Latest version:** You get the most recent version of the program.
- **Legality:** You get a license for unlimited use of the software. Your name will be stamped in the EXE file and will appear in the About box of the program. If you do not register, your license to use the software expires after 30 days.
- **Support:** Free Technical Support is available to the registered users. Please see the Support Policy, described later in this topic.
- **Complete desktop restoration:** In the registered version, you can use the "Restore desktop on startup" option (menu Options--General Desktop Options). If you do that, all your folders, topics and text files which were open at the time of last program exit will be restored in the same positions.
- **Spelling-Checker option:** You can order an English Spelling-Checker (£10) which works only with the registered version of WhizNotes, WhizFolders or HelpHikes Pro. This Spelling-Checker includes both American and British dictionaries. If enough users are interested in another language dictionary, it can be made available.
- **The Registration Reminder screen is removed:** This makes it more convenient to start up the program from the WZN Shortcuts created on the desktop.
- **Additional utilities:** The following additional utilities are available with the registered version:
 - **FIXWZN.EXE** - My command-line utility to fix the WZN files, if corrupted due to any malfunction in the system.
 - **CONVINC.EXE** - My command-line help generation utility to convert the #define statements in INC file to any other language based on a template that you supply.
 - **WZLOCATE.EXE** - My command-line help generation utility to locate the Hex error positions reported by the help compiler for the input RTF file.
- **Limits removed:** The limit on the number of topics in a folder is removed so you can add hundreds of topics to a folder. The limit on the number of help topics in a folder is also removed so you can create help files with hundreds and even thousands of topics.
- **Unregistered message removed:** The message "This help file has been created with unregistered copy of WhizNotes" will not appear when you generate help files with the Registered Version of WhizNotes or HelpHikes Pro. You get the **license for unlimited commercial usage of the program** to generate the RTF for creating help files. You also get the license to distribute any WZN files that you create using the program.

- **Advanced Help Generation features:** In the registered version of WhizNotes and HelpHikes Pro, you get some **advanced** help authoring features not available in this evaluation copy of WhizNotes. Unless you register, I don't want to add to my support costs by answering questions for these features. This is based on my past support experience for WhizNotes. These features are Help Macros, Startup Help Macros, RtfMacros to create your own HHML statements, Browsing sequences, and customizing the Help Project (HPJ) file. But, all the documentation for these features is available in this evaluation version's online help. You can read the help topics for these features to know more about them.

In HelpHikes Pro, you get many more help generation features. For details, please see [HelpHikes Pro, additional features](#).

- **Additional samples:** You get additional examples for the above advanced features in the sample WZN files given with the registered version. For instance, in WhizNotes, a sample file demonstrates how to customize the RTF header for using multiple embedded fonts in the help text. In HelpHikes Pro, this has been made much easier so that you don't have to code an RTF header to do that. HelpHikes Pro contains an additional sample file to demonstrate its Windows 95 style help generation capabilities.
- **Updates and Upgrades:** Please see the Support Policy, described later in this topic.
- Registered users can subscribe to a FREE email newsletter informing of updates and tips for using these programs.

Support policy:

Technical support by electronic mail or through Web page: Only Online Support is available for my programs through my [web page](#) or by electronic mail. If you don't have access to these facilities, it's recommended that you get them as soon as possible. Otherwise, you are going to miss on the new features and documents which will be released by intermediate updates or upgrades and will only be available on the web page.

Updates and Upgrades: Smaller improvements and minor bug-fixes will only be available through my [web page](#), free of cost. No disk will be mailed except for major bug-fixes, if any. Major improvements and add-on programs will be available at a discounted, upgrade price to the registered users either by email registration or through a disk ordering service.

Site Licenses:

Site License discounts are available. Please see [Site License Prices](#).

Please Register and support innovative development:

If you like WhizNotes, please register one of the programs WhizNotes, WhizFolders or HelpHikes Pro, depending on your needs. Send your suggestions and tips on how you use these programs and participate in this exciting, ongoing development that has evolved based on the needs of many *creative users* like you.

Thank you,

Sanjay Kanade

Author of WhizFolders, WhizNotes and HelpHikes Pro

Email: skanade@superb.net

Also see:

[Magazine quotes on WhizNotes](#)
[Ordering details](#)

[Back](#)

Before you begin, why not see a demo of help generation! Just open the sample folder SAMPLHLP.WZN. Now, to compile it, just press the 'H' button on the toolbar or click on the menu **Tools-Create Help**, then press OK on the dialog that comes up. It's as simple as that.

If you don't have a help compiler properly installed on your system, you'll get a suitable error message. In that case, please consult the topic [Setting up the Help Compilation environment](#).

To know the details, please take a look at all the topics in this sample folder and consult the online help as you go. For more details on the demo, please see the topics [Sample WZN file to demonstrate Help Generation](#) and [More about the demo](#).

Startup Tips and Tricks

If you're using Windows 95, you must have become used to its convenient *document-centric* approach. The idea is to never think of what program we need to run. Instead, we simply double-click on a document shortcut and it starts the proper program for us. However, the program also needs to support this start up. In particular, if the program is a multi-window application like WhizNotes and it's already running, it should open the document in a new window.

WhizNotes supports a document-centric startup on Windows 95 and also on Windows 3.1. Not only can you open a WhizFolder document by double-clicking on it in the File Manager or Explorer, you can also create shortcuts on your desktop for the most used WZN files. Here's how to do that.

Creating a desktop shortcut on Windows 95:

Start up the Explorer and go to the directory where the required WZN file is present. Click on the WZN file with the right mouse button and drag it to the desktop with the right mouse button. When you leave the mouse button, a menu appears where you can select 'Create shortcut here' operation. This creates a desktop shortcut for that WZN file so that whenever you double-click on the shortcut, the file will be opened for you in WhizNotes.

Creating a desktop shortcut on Windows 3.1:

Some of you might think, the convenience of opening a file from a desktop shortcut is only available on Windows 95. Be surprised! You can create shortcuts in Windows 3.1 too but they can only be created in a Program Manager group window. *This also gives you a nice way to give a long name to WZN files on Windows 3.1, at least on their desktop shortcut.*

To do that, simply open the desired Program Manager group where you want to create the shortcut. Then from the File Manager, drag the WZN file with the left mouse button on to that Program Manager group. This will create a shortcut for you with which you can always open the WhizFolder document by double-clicking on it. After creating the shortcut, you can rename it to a long name.

Using WhizNotes as a Help Authoring System

Basics of Help generation:

[Basics of Help generation](#)
[Setting up the Help Compilation environment](#)
[Terminology of help generation](#)
[Sample WZN file to demonstrate Help Generation](#)

HelpHikes Markup Language (HHML):

[What is HHML?](#)
[HHML: List of statements supported by WhizNotes](#)
[WhizMenu for quick entry of canned text and HHML statements](#)

The magic of Hypertext links:

[Inserting a jump to another topic](#)
[All kinds of Jumps & References to another topic](#)
[Specifying jumps & references to other help files](#)
[Browsing sequences of topics](#)
[Defining keywords for topics](#)
[Keywords, macros, and other characteristics of topics](#)

Bitmaps, hotspots, and Hypergraphics:

[How to insert bitmaps into your help files](#)
[Bitmaps as hotspots](#)
[Bitmaps as hypergraphics](#)

Character/paragraph properties:

[Making your text bold, italic, etc.](#)
[Changing styles of topic titles](#)
[Making bulleted and numeric lists](#)
[Indenting text](#)
[Starting a new line within a paragraph](#)
[Displaying topic title in a non-scrolling region](#)
[Displaying text in tables](#)

Help Macros:

[Inserting help macros](#)
[Defining startup help macros](#)

Programmer's corner:

[Features for context sensitive help](#)
[Conditional help generation](#)
[Hints on testing the help file](#)

The power to customize:

[Creating your own custom statements with RtfMacros](#)
[Modifying RTF and HPJ files with another tool](#)
[Conditional help generation](#)
[The power of the HelpTopic paradigm](#)
[Inserting direct RTF code](#)
[Global options to control help generation in general](#)
[Customizing Help Project \(HPJ\) file](#)
[Less often used INI flags](#)

Hints on testing the help file

HelpHikes Pro, additional features

This topic describes HelpHikes Pro, a more advanced Help Authoring System so that you can make a decision whether you want to upgrade to HelpHikes Pro. Note that HelpHikes Pro is not a shareware program and no evaluation version is available for download.

This help file that you are reading has been created with HelpHikes Pro. Although, HelpHikes Pro fully supports Windows 95 style help creation, I didn't use it here because my help file targets both Windows 3.1 and Windows 95 systems. To see a real example of Windows 95 help, please access my web page to get a sample Win95 style demo file created with HelpHikes Pro.

How HelpHikes Pro came into being:

Once you start creating Windows help files with WhizNotes and become experienced, you will want more features. The idea of WhizNotes encourages creativity. Its customization features allow you to understand help creation from a different perspective. This new found creativity is sure to generate more ideas for enhancements. In fact, it did for myself and for many many WhizNotes users. I got so many ideas from the users and from my own research that it was not possible to put all of these into WhizNotes as it is more of a combination of a note taking program and a help generator. There are also some legal and marketing issues involved in supporting this decision to stop adding more help generation features to WhizNotes.

Hence, a new program **HelpHikes Pro** was born. **HelpHikes Pro** is a professional extension of WhizNotes' Help Authoring features. The user interface is very similar to WhizNotes so that you waste no time in learning it. At the same time, you get a host of new and more powerful features. So if you are really into help authoring as a professional, you should upgrade to **HelpHikes Pro**. For registered users of WhizNotes, very attractive upgrade prices are available.

Note that WhizNotes is not going to get any new help generation features as it is a dual purpose program. It will remain at this current feature level for a much longer time. It's **HelpHikes Pro** which will evolve more rapidly to fill the needs of the users. With **HelpHikes Pro**, you get all the features that WhizNotes supports, plus many more, including the support for the latest Windows 95 help. For a quick list of the major additional features available in **HelpHikes Pro**, please see the following list. You can also access my web page for latest details on HelpHikes Pro. It's possible that I might put the whole help file for HelpHikes Pro to be downloaded so that you can look at the whole documentation in detail to evaluate whether you want to upgrade to HelpHikes Pro.

Smart build:

HelpHikes Pro provides a unique option **Smart build** which is similar to **make** or **IDE project** concepts known to C programmers. If you select this option and make a change in your help topics, it finds out whether it needs to build RTF for the whole help from the beginning or just create RTF for the changed topic. This feature will save you lots of time when authoring help files.

Support for secondary windows:

Secondary windows are important elements of serious online help. HelpHikes Pro supports secondary windows along with all the new features available in the new HCW compiler and the Windows 95 help. Secondary windows can be defined by using the

DefineWin tools in ~RtfMacro topics. Jumps to secondary windows are provided with JumpWin statements. You can also assign a window to a topic through its HelpTopic statements so that all jumps to that topic are automatically assumed to be JumpWin statements, even the jumps from keyword indexes. It is unlikely that you will find this level of object-oriented abstraction in any other authoring tool.

Many more HHML statements and tools:

HHML really belongs to HelpHikes Pro and it defines a complete set of statements and tools, allowing you to define fonts, sizes, colors, margins, spacing, non-scrolling regions, indent values, conversion tools from inches to twips, etc. You can even change the default font and size for a whole topic, for example, when displaying an order form to be printed or when displaying a topic in small font size in a secondary window.

Also to make your RtfMacros more powerful there are HHML statements to get the context string for a topic, visible name of a topic, etc.

More customization features:

The real power of RtfMacros feature is evident in HelpHikes Pro where you can use other RtfMacros in the definition of an RtfMacro recursively. This way, you can define text styles using certain fonts, colors, point sizes, etc. The defined text style becomes an HHML statement which you can use in the definition of another style! So, if your boss doesn't like some style, you can change it in only one place where it's defined.

To give you even more power, an RtfLinkMacro feature allows you to make your own Jump statements for any degree of customization. You will be amazed with the new jump statements created for you in the sample help.

Full support for the Windows 95 help features

Creating .CNT files for Windows 95 contents is easy with HelpHikes Pro. Just create a topic ~CntFileContents and drag and drop jumps on it. This feature is independent of the compiler so you can create .CNT contents even for the help files compiled with old DOS based help compilers.

HelpHikes Pro fully supports all the aspects of using the new HCW compiler to create the Windows 95 help. You can access my web page and download a sample WZN file along with the help created from it using the new HCW compiler.

Expect more features to be added:

As I said, in addition to myself, many creative users like you are generating ideas for enhancements. You can expect more features and add-on programs in future for HelpHikes Pro. And there's always going to be an attractive upgrade price for the registered users.

Other programs in the WhizFolder family

History:

Want to see a little bit of history on why I chose to create three programs? See [Why three programs?](#)

WhizFolders - A Hypertext Organizer for Windows

WhizFolders has everything that WhizNotes has minus the help or RTF generation capabilities. It is intended for those users who like WhizNotes but are not interested in help generation. In other words, If you never use the menu items **Tools - Create Help** or **Tools - Export selected topics as non-help RTF** then WhizFolders is enough for your use.

Hence, WhizFolders is mainly a note taking program and will grow to address the general hypertext organization and note taking needs of the users.

HelpHikes Pro - A Help Authoring System for Windows 95

HelpHikes Pro is a more advanced help authoring system than WhizNotes and hence costs more. Note that it can run on both Windows 3.1 and Windows 95 and can produce help files for both platforms. It also fully supports all the features of Windows 95 help creation using the new HCW compiler. It has more features than WhizNotes even for Windows 3.1 help creation. For details on its additional features, please see [HelpHikes Pro, additional features](#).

Spelling-Checker for above programs:

A Spelling-Checker is available as an optional add-on product for all the three programs: WhizNotes, WhizFolders and HelpHikes Pro. One Spelling-Checker supports all of them and can be shared by these programs. Currently only an English spelling-checker for American and British English is available. If you want to buy a different language, please send an email message to the author. If enough users request a particular language, it can be made available sooner.

Locate self in parent list

This convenient operation is available on the Search menu when you are working in a topic window. It quickly switches to the parent WhizFolder of this topic and brings this topic into view in the list of topics, selecting it at the same time.

It can be very useful in many situations. For example, let us say you are opening your topics in a sequence to check them. Suddenly you feel a need to search for some text in the parent WhizFolder. After performing this search, you lose your original position in the WhizFolder to continue checking topics. In this case, simply select the last topic window and perform this operation. You are back to the last topic's position in the list.

Some text which you need to use often. For example, outline of a form or a letter. The word 'Canned' here means 'packaged'. Canned text is your own package of text for your own use.

HelpHikes Pro is a Help Authoring System by the same author and has many more features than WhizNotes. For more details, please see [HelpHikes Pro, additional features.](#)

HHML means "HelpHikes Markup Language." A [**~jump...**] statement is an example of an HHML statement. Many of HHML features have been designed for creating Windows help files. But you can also use HHML to create RTF files for use in your word processor.

The term HHML comes from [HelpHikes Pro](#) which defines the complete keyword set for the language. WhizNotes uses only a subset of HHML. See [What is HHML?](#). Also, a complete list of HHML statements is given in [HHML: List of statements supported by WhizNotes](#).

Who needs to create Help files?

Windows help was earlier *exclusively* used for providing online program documentation. Not any more. WhizNotes makes help generation incredibly simple so that you can make help files for any type of information whether it is a collection of your recipes or your weekly reports. Also, these help files can be distributed to other Windows users who can immediately benefit from your notes and ideas. Anyone can use them on any system where Windows 3.1 or Windows 95 is installed. No special software is required to look at these help files. With WhizNotes, help files become so simple to create that they assume an important role. In their natural hypertext form, they become a perfect vehicle for sharing information between people.

Ordering details

Order any of the three programs!

You get a choice of ordering the registered version of any of the three similar programs, depending on your needs. These programs are WhizNotes, WhizFolders and HelpHikes Pro. If you have seen what WhizNotes can do for you, you should also see [what is different in WhizFolders or HelpHikes Pro.](#)

Ordering information and prices:

Please see [Ordering information and prices.](#)

Ordering information and upgrade prices for registered users:

If you are a registered user of any program in the WhizFolder family, please see [Upgrade prices for registered users.](#)

Site Licenses:

Please see [Site License Prices.](#)

Money-back guarantee:

All the programs are backed by the author's unconditional 30 days money back guarantee.

The following form is also available in the file ORDER.TXT.

Programs by Sanjay Kanade / Ensign Systems Order Form

Important: If you are ordering by mail, e-mail, or fax, please completely fill out the form on the next page and send it in. If you are sending a cheque or money order, please make sure that payment is made in UK Sterling drawn on a UK Bank, by Eurocheque drawn in Sterling, or in \$US drawn on a US bank.

TECHNICAL SUPPORT: Please note that technical support is NOT available from Ensign Systems. Only Online Support is available for these programs through the Author's web page or by electronic mail. For any technical help, first visit the author's web site to see if your question is answered there. If not, please contact the author, Sanjay Kanade, by email on one of the following addresses:

"Sanjay Kanade" <skanade@superb.net>
Author's web site: <http://skanade.simplenet.com/>

You may mail, fax, e-mail, or phone in your order. Please send the completed registration form, along with payment to Ensign Systems at:

ENSIGN SYSTEMS	TELEPHONE 0181-778-2871
21 TREDOWN ROAD	FAX 0181-776-8477
SYDENHAM	EMAIL ENsignpub@aol.com
LONDON SE26 5QQ	or ensignpub@clara.net
UNITED KINGDOM	

UPGRADE PRICES: Please see the file UPGRADE.TXT or refer to the author's web page.

DISCOUNTS: If ordering 3 or more copies, please see the Site License table on the last page.

(Your Name)

(Your Company)

(Street1)

(Street2)

(Town)

(County)

(Postcode)

(Telephone)

(Fax)

(Country if outside UK)

(e-mail address)

DISCOUNTS: If ordering 3 or more copies, please see the Site License table on the last page.

QUANTITY	ITEM	PRICE EACH	TOTAL
(All prices in £Sterling)			
A.	WhizFolders	£26	£
B.	WhizNotes	£32	£
C.	HelpHikes Pro	£49	£
D.	Spelling-Checker for A, B, or C (American & British English)	£10	£
	Shipping		£2.00
	(Shipping £2.00 for your entire order, whether you order 1 or 100 items.)		
		TOTAL: £Sterling	=====

For these programs, orders are ONLY sent on 3.5" diskettes.

I am paying by:

() CHEQUE () VISA CARD () MASTERCARD () DELTA () EUROCARD

Number _____ - _____ - _____ - _____ Expiry date ____ - ____

(Signature)

(Date)

OPTIONAL INFORMATION:

We would highly appreciate if you tell us where did you first hear about these programs. Thanks for your cooperation.

[] WhizNotes shareware version

[] BBS/Online service - which one? _____

[] Internet - which site? _____

Author's []

Ensign's []

other? _____

[] If you found the site by search, which Search Engine did you use? _____

[] Magazine _____

[] Friend - _____

Do you have a cd-rom? [] yes [] no

The evaluation version of these programs, if available, may be downloaded from the following locations. At the time of this writing, only WhizNotes has an evaluation version for download.

Ensign's web site- <http://www.ensignpub.com/>

Author's web site- <http://skanade.simplenet.com/>

SITE LICENSE PRICING:

 If buying 3 or more copies, you get a discount as per the following table. Note that the Spelling-Checker is included in these prices.

A Site License includes the following:

- * One diskette and documentation set for the program.
- * One diskette containing the English Spelling-Checker.
- * An agreement specifying the number of copies that can be made as per the license agreement.

To order, please get the total price from the following table and use that price in the above order form. Note that many columns include a fixed base price and a per copy price.

Number of users	WhizNotes + Spell-Check	WhizFolders + Spell-Check	HelpHikes Pro + Spell-Check
1- 2	\$65 each	\$55 each	\$90 each
3- 5	\$55 each	\$45 each	\$75 each
6- 15	\$55+\$40 each	\$45+\$35 each	\$75+\$55 each
16- 25	\$205+\$30 each	\$175+\$27 each	\$225+\$45 each
26- 50	\$475+\$20 each	\$435+\$17 each	\$705+\$26 each
51-100	\$885+\$12 each	\$745+\$11 each	\$1105+\$18 each
101-more	\$1395+\$ 7 each	\$1250+\$ 6 each	\$2000+\$ 9 each

For example, the total price for WhizFolders, 25 users, would be (using the fourth row of the table above):

$$\$175 + (\$27 * 25) \text{ users} = \$850 \text{ only (almost 40\% discount)}$$

UK purchasers should convert the US dollar prices above to £'s Sterling by dividing by 1.60. This rate may vary according to currency fluctuations and UK purchasers are advised to contact Ensign on 0181-778-2871 prior to ordering to get a quotation at the prevailing rate. We reserve the right to charge site licences at the prevailing exchange rate.

Release note

I am happy to inform you that in this version, the right-click menu tool WhizMenu has been made available so that you can try it for smart template insertions. For more details, please see [WhizMenu for quick entry of canned text and HHML statements](#). You can even use it for HTML authoring. In fact, I have included a sample menu file for that purpose.

You will also see many improvements in the user interface, such as a completely revamped right-click speed menu on the topic list. Please purchase the licensed version as soon as possible for unlimited use of the program.

Sanjay Kanade
Author of WhizFolders, WhizNotes and HelpHikes Pro
Email: skanade@superb.net

<http://skanade.simplenet.com/>

Web sites

For more information on WhizNotes, please visit the following web sites.

For the latest information on how to order WhizNotes, please visit the web site of the Ensign Systems:

<http://www.ensignpub.com/>

For the latest support information on WhizNotes, please visit the author's web page:

<http://skanade.simplenet.com/>

This topic on the right is the topic "Getting started" from Online Help. It always remains on top so that when you carry out the actions suggested, they remain visible for your reference.

You can remove this topic from view by minimizing it. If you close it, you can come back to it later from the menu **Help-Quick Start** or from the "Introduction" section in the Help Contents.

Your License for Trial Use

This is an evaluation copy of WhizNotes. As a non-registered user of this software, you are granted a limited license to use this copy for the express purpose of determining whether WhizNotes is suitable for your needs. At the end of a 30-day trial period, you should either purchase the registered version or discontinue using WhizNotes.

You get a choice of [ordering the registered version](#) of one of the three similar programs: WhizNotes, WhizFolders and HelpHikes Pro. To get more details on WhizFolders or HelpHikes Pro programs, please see [what is different in WhizFolders or HelpHikes Pro](#).

Also see:

[What you get when you Register](#)
[Ordering details](#)

Ordering information and prices

Ordering a disk:

Program name	Price (per copy)	Shipping charges (any number of copies)
WhizNotes	£32	£2
<u>WhizFolders</u>	£26	£2
<u>HelpHikes Pro</u>	£49	£2
<u>English Spelling-Checker</u> Works with any of the above programs	£10	£0.00 (If ordered with one of the above programs) £2 (if ordered separately)

Payments by credit card:

To order, please call **Ensign Systems** at:

TELEPHONE: 0181-778-2871

FAX: 0181-776-8477

EMAIL: ENsignpub@aol.com

or ensignpub@clara.net

Website- <http://www.ensignpub.com/>

ENSIGN SYSTEMS
21 TREDOWN ROAD
SYDENHAM
LONDON SE26 5QQ
UNITED KINGDOM

VISA, MASTERCARD, DELTA and EUROCARD are accepted. If you are sending a cheque or money order, please make sure that payment is made in UK Sterling drawn on a UK Bank, by Eurocheque drawn in Sterling, or in \$US drawn on a US bank.

When ordering 3 or more copies you can order a [Site License and get discounts.](#)

Order form:

You can also use an Order Form when paying by a credit card or by a check. Please print the order form from the Registration Reminder Screen (menu About-Registration Reminder). You will also find this order form in the file ORDER.TXT or in the help topic [Ensign Systems Order Form.](#) You can print it from the help topic, but it's recommended that for proper page breaks, you either print it from the initial Registration Reminder screen or open the file ORDER.TXT in WhizNotes as a text file and print it.

Upgrade prices for registered users

Special upgrade prices are available for the registered users of WhizNotes, WhizFolders or HelpHikes Pro to get another program from the WhizFolder family. For example, if you have a licensed copy of WhizFolders or WhizNotes (old or new) and want to upgrade to HelpHikes Pro, you can do so at an upgrade price.

Ordering an "Upgrade disk":

WARNING: Note that when you use an upgrade disk for the *first time*, it will look for the presence of a registered version of WhizNotes(old or new), WhizFolders or HelpHikes Pro on your system. It won't allow an installation unless it finds a registered version installed. After the first *successful* installation, the disk will be modified so that it can be used anywhere, on any PC.

Program name	Upgrade Price (per copy)	Shipping charges (any number of copies)
WhizNotes	£19	£2
<u>WhizFolders</u> Upgrade	£16	£2
<u>HelpHikes Pro</u> Upgrade	£26	£2
<u>English Spelling-Checker</u> Works with any of the above programs	£10	£0.00 (If ordered with one of the above programs) £2 (if ordered separately)

Payments by credit card:

To order, please call **Ensign Systems** at one of the following numbers and mention that you are ordering an "Upgrade". They might also ask you for approximate month and year when you purchased the previous registered version.

TELEPHONE: 0181-778-2871

FAX: 0181-776-8477

EMAIL: ENsignpub@aol.com

or ensignpub@clara.net

Website- <http://www.ensignpub.com/>

ENSIGN SYSTEMS
21 TREDOWN ROAD
SYDENHAM
LONDON SE26 5QQ
UNITED KINGDOM

VISA, MASTERCARD, DELTA and EUROCARD are accepted. If you are sending a cheque or money order, please make sure that payment is made in UK Sterling drawn on a UK Bank, by Eurocheque drawn in Sterling, or in \$US drawn on a US bank.

Order form:

You can also use an Upgrade Order Form when paying by a credit card or by a check. You will also find this order form in the file UPGRADE.TXT or in the help topic [Upgrade order form](#). You can print it from the help topic, but it's recommended that for proper page breaks, you open the file UPGRADE.TXT in WhizNotes as a text file and print it.

The following form is also available in the file UPGRADE.TXT.

| Programs by Sanjay Kanade / Ensign Order Form |
UPGRADES

Special upgrade prices are available for the registered users of WhizNotes, WhizFolders or HelpHikes Pro to get another program from the WhizFolder family. For example, if you have a licensed copy of WhizFolders or WhizNotes (old or new) and want to upgrade to HelpHikes Pro, you can do so at an upgrade price.

WARNING: Note that this is an order form for an UPGRADE DISK. When you use an upgrade disk for the "first time", it will look for the presence of a registered version on your system. It won't allow an installation unless it finds a registered version installed. After the first successful installation, the disk will be modified so that it can be used anywhere. So, please order an upgrade disk only if you have your registered version properly installed. If not, you should either order a regular disk at FULL price or contact Ensign for Online registration so that we can verify your upgrade eligibility and can give you an upgrade registration by email.

Important: If you are ordering by mail, e-mail, or fax, please completely fill out the form on the next page and send it in.

TECHNICAL SUPPORT: Please note that technical support is NOT available from Ensign Systems. Only Online Support is available for these programs through the Author's web page or by electronic mail. For any technical help, first visit the author's web site to see if your question is answered there. If not, please contact the author, Sanjay Kanade, by email on one of the following addresses:

"Sanjay Kanade" <skanade@superb.net>
Author's web site: <http://skanade.simplenet.com/>

You may mail, fax, e-mail, or phone in your order to:

ENSIGN SYSTEMS	TELEPHONE 0181-778-2871
21 TREDOWN ROAD	FAX 0181-776-8477
SYDENHAM	EMAIL ENsignpub@aol.com
LONDON SE26 5QQ	or ensignpub@clara.net
UNITED KINGDOM	

(Your Name)

(Your Company)

(Street1)

(Street2)

(Town) (county) (Postcode)

(Telephone) (Fax) (Country if outside UK)

(e-mail address)

QUANTITY	ITEM	UPGRADE PRICE EACH	TOTAL
(All prices in £Sterling)			
A.	WhizFolders (list price US\$40)	£16.00	
B.	WhizNotes (list price US\$50)	£19.00	
C.	HelpHikes Pro (list price US\$75)	£26.00	
D.	Spelling-Checker for A, B, or C (American & British English)	£10.00	
	Shipping		£2.00
(Shipping £2.00 for your entire order, whether you order 1 or 100 items.)			
	TOTAL:	£Sterling	=====

For these programs, orders are ONLY sent on 3.5" diskettes.

I am paying by:

CHEQUE VISA CARD MASTERCARD DELTA
 EUROCARD

Number _____ - _____ - _____ - _____ Expiry date ____ - ____

(Signature)

(Date)

OPTIONAL INFORMATION:

We would highly appreciate if you tell us where did you first hear about these programs. Thanks for your cooperation.

WhizNotes shareware version

BBS/Online service - which one? _____

Internet - which site?

Author's

Ensign' s

other? _____

If you found the site by search, which Search Engine did you use? _____

Friend retail store package CD-ROM _____

magazine - which one? _____

other - _____

Do you have a cd-rom? yes no

The evaluation version of these programs, if available, may be downloaded from the following locations. At the time of this writing, only WhizNotes has an evaluation version for download.

Ensign's web site- <http://www.ensignpub.com/>

Author's web site- <http://skanade.simplenet.com/>

Why three programs?

Here's a little history on why I chose to create three different programs.

As I had mentioned in the earlier versions of WhizNotes, "The use of this program is only limited by your imagination!" It was confirmed by the kind and volume of email that I received from the excited users. The messages ranged from "how they are using WhizNotes," "how much they appreciate it" to "what if this feature could be added," "in which direction they would like the program to grow," and so on. I really appreciate all the feedback that I received.

Based on this volume of messages, I could classify the users of WhizNotes into three types of users:

1. **"Note taking and organization" Users:** Users who used it for notes and text organization, outlining and random information management. These users weren't interested in help or RTF generation and wanted the program to grow to support more organizational features.
2. **Help and HyperText Enthusiasts:** Users who were thrilled with the new found creativity that allowed them to create Help and Hypertext documents easily and effectively. Many users of this type created help for the first time with WhizNotes. They were happy with what WhizNotes could do in the area of help generation. In fact, WhizNotes had more help generation features than they cared for.
3. **Serious Help Authors and Programmers:** These were the users who used every imaginable feature provided by WhizNotes for help generation and even developed an interest in RTF coding with RtfMacros and what they could achieve with it. Some of them stretched the limits of WhizNotes 2.13m by creating help files with thousands of topics in them. They wanted more features to handle stuff like heavy customization of the topic titles and appearance, embedded fonts, colors, secondary windows, recursive RTF macros, ability to handle thousands of topics, smart building of RTF to save time on RTF generation, etc. Moreover, with the new Windows 95 help and the new help compiler, HCW, another dimension was added to the help authoring with a vast scope for improvement.

As you can see, a single program, WhizNotes, could not continue to serve all these types of users. Hence, I decided to develop two new programs. Here is how I believe the three programs are meant to satisfy three types of users:

1. **WhizFolders for the "Note taking and organization" Users:** I created WhizFolders for these users. WhizFolders has everything that this WhizNotes has, minus the RTF and help generation features. Also, WhizFolders is not a shareware program. There is no evaluation version available for download. But, as it is a subset of WhizNotes, you can evaluate all its features in this evaluation copy of WhizNotes. Currently, WhizFolders costs less than WhizNotes as it is a subset. But, it's eventually going to get beyond WhizNotes or HelpHikes Pro in the general organizational features. Since, now, a line has been drawn to separate this WhizFolders and give it a definite direction, it is going to create its own following of users and will grow rapidly. If you are a user in this category, you should go for WhizFolders.
2. **WhizNotes for the Help and HyperText Enthusiasts:** WhizNotes will continue to serve this category of users and will remain a shareware program. But that also means that the program has almost matured and there are limits to its growth. If you are a user in this category, you can start with WhizNotes and upgrade to HelpHikes Pro when you

become more adept at help authoring. Also note that if you are using the Export RTF feature in WhizNotes, you won't find it in WhizFolders. It is available only in WhizNotes or HelpHikes Pro.

3. **HelpHikes for the serious Help Authors and Programmers:** I created HelpHikes Pro for this category of users. It addresses all those requirements mentioned earlier for these users, such as, recursive RtfMacros, secondary Windows, embedded fonts, innovative smart build, etc. Also, this is not a shareware program and there is no evaluation version for download. But, you can evaluate many of its help generation features in this evaluation copy of WhizNotes. The user interface remains exactly same as WhizNotes! To see what additional features it supports, please see [HelpHikes Pro, additional features.](#)

HelpHikes Pro will continue to grow in the area of help generation. Its current RtfLinkMacro feature gives you unprecedented power to create your own types of hypertext link statements. Also, it provides full support for both Windows 3.1 and Windows 95 help. If you are a serious help author, consider buying or upgrading to HelpHikes Pro. You won't find an equivalent of its innovative Smart Build feature in another tool.

Please check the program's speed

Please open one of the folders installed with the program, such as SAMPLHLP.WZN. Even if you are using a 386 system, it should not take more than a few seconds to open or close it. You can expect a similar speed for opening and closing other folders containing about 50 topics. If you are satisfied with the speed of opening of such small folders, then you need not read the rest of this topic. You are OK.

If you find that opening and closing a small folder is taking an unusually long time, there is something wrong with your Windows configuration. You need to perform some experiments to improve the speed. This is important. Please don't ignore it. For example, one of the users found that a folder with about 50 topics was taking about 60 seconds to open on his system. After he performed the following steps, it came to less than 3 seconds!

Possible reasons for the degradation of speed: If you find such slow speeds then most probably you are using double-spaced drives. Some users even install Windows on compressed space which might slow things down. WhizNotes, by default, creates some temporary directories under the directory used by Windows for temporary files. For some systems, creation of these directories becomes very slow. There may be several reasons for this. There may be too much junk in the temp directory or it might be situated on a highly fragmented compressed drive.

If you are confident about Windows installation procedures, you should try to move this temp directory to an uncompressed drive. If you don't want to do it, WhizNotes provides another solution as given next.

Telling the program to create temporary files elsewhere

You will find a file **WHIZNOTE.INI** in your Windows directory. Please open it using a text editor or open it in WhizNotes as a text file. Add the following lines to it, taking care not to add a line if it is already there.

```
[RareUsageFlags]
UseWorkTempDir=y
```

Once you do that, stop and restart the program WhizNotes. It will now use the working directory for creation of temporary files. By default, the working directory is same as the program's own directory. You can confirm it by the existence of a subdirectory WHIZTEMP only when the program is running.

This should fix the problem. If the speed doesn't improve, check if your program WhizNotes itself is installed on a compressed drive, you can move it to an uncompressed area and run Setup in the new location to perform the operation "Reinstall after a move". If you don't want to do that, consider changing the working directory to a directory on uncompressed space. You can do that by Program Manager's properties for the WhizNotes icon or by changing the "Start in" directory on the Windows 95 properties for the WhizNotes shortcut.

Site License Prices

If buying 3 or more copies, you get a discount as per the following table. Note that the Spelling-Checker is included in these prices.

A Site License includes the following:

- One diskette and documentation set for the program.
- One diskette containing the English Spelling-Checker.
- An agreement specifying the number of copies that can be made as per the license agreement.

To order a Site License, please use the table of discounts at the end of main order form in the file ORDER.TXT.

Magazine quotes on WhizNotes

Here are some interesting quotes on the earlier versions of WhizNotes:

"It's a slick Windows Help authoring tool. Generating a help file is about as painless as it can be. You don't have to think about context strings, and you never so much as glimpse the project file. Nor do you need to know anything about the help compiler--other than to make sure it's installed in your DOS path."

Mike Lewis

"Bypass Help Hassles with WhizNotes"
DBAdvisor, July 95

"Yes, there are many, many text editors/word processors available but, all they really and truly offer are "bells and whistles" to make your presentations look great. But what do they do to help you create its content? After all, the content is the MOST important part of your work. These other programs focus on the "finished" product but, do not do very much to help you in the "creation" process. This is the area where WhizNotes excels!! It helps you get all those bits and pieces of data floating about in the "sea of your mind" and nets them together into a useable and coherent form!"

Mythdancer's shareware review

June 1995

"WhizNotes 2.1 is the easiest way to brainstorm in Windows without paying for a secretary. As a brainstorming tool and idea organizer, WhizNotes is everything its name implies."

Barry Brenesal

"Organizing your ideas is a breeze with WhizNotes 2.1"
Computer Shopper, May 95

"You'll be a whiz at managing information with this notepad utility. Create "topics"--as many as you want--and group them in parent topics (WhizNotes). Then open multiple WhizNotes and conduct searches. Using the Microsoft Help Compiler, you can even turn WhizNote files into a Help file."

Julie R. Blumenfeld

"Superior Shareware"

Windows Magazine, Appeared twice, in 94 and 95.

"Super-notepad helps you organize your tasks, take notes, prepare documents and reorganize your thoughts."

Compuserve magazine, May 94

"WhizNotes combines elements of outlining, text editing, and personal information management to produce a valuable research and organizational tool."

Charles Bermant

"Cover Story: Windows Software Gems"
Bay Area Computer Currents, Mar 94

"To me, WhizNotes is the epitome of what a good Windows program should be. You can use it without referencing a manual. As your familiarity with its features grows, additional uses spring naturally to mind."

Keith L. Miller

"Law Office Technology"

Minnesota Trial Lawyers Magazine, Summer 94

"Realistically, WhizNotes is a clever implementation of a smart idea, and provides a solid way to plan and organize a project. I, for one, will use this program for my next major

project."

Charles Bermant

"Windows Chronicles"

Bay Area Computer Currents, Nov 93

Locate GlobalHelpOptions Definition

Quickly locates the GlobalHelpOptions statement if you coded it.

This action is also available on the Right-click menu on the topic list under a submenu called "Search For Help Attribute...."

Also see:

[Getting WhizFolder details by right-click](#)

Always keep on top

Check this menu item to keep the WhizNotes always on top of all Windows. Uncheck it to make it normal.

Browse next topic

Click on this menu item to open the next topic in the topic list. You can also press Alt-Down keys for the same operation.

Browse previous topic

Click on this menu item to open the previous topic in the topic list. You can also press Alt-Up keys for the same operation.

Save topics to text file

You can use the "Save topics" operation on the File menu to save all the selected topics together in a text file. This is similar to the export operation but is simpler as it does not contain the delimiter, word-break and other advanced export options.

Also see:

[Exporting selected topics to a text file](#)

Windows 95 help feature

The feature described on the right is only available in the help files for Windows 95 or a later version of Windows. To use this feature, you need to use the compiler HCW, Microsoft Help Authoring Workshop.

