WhereSIT

The *StuffIt* Archive Searching Utility for the Apple Macintosh

---

Written by Robert S. T. Gibson

Written in *THINK's LightspeedC*™

Chapter 1 : Introduction

What is *WhereSIT*?

WhereSIT is an application which functions similarly to a file finder (for example *Find File*, by David Goldsmith, distributed with the Apple System disks). It searches for a *StuffIt* archive in the specified directory and scans it for a search key name. It lists all files within an archive which it finds that contain the key word in their names, and allows you to view different information about the file. You can also move the file to the desktop for further manipulation. When this is done, the file can be returned to it's original directory by the Finder's Put Away command.

What is *StuffIt*?

The scope of this guide does not cover the details of *StuffIt*. All that can be said is a short, general overview of its capabilities.

StuffIt, *the* file compression utility for the Macintosh, is a file archiving utility which allows you to group files together, compress the files, encript the files, and segment the groups for saving on backup disks. The gathering of these files allows faster and much simpler transmission of data by modem, and also saves storage space if used by files you do not use frequently. Its encryption protects your data from prying eyes.

You should be familiar with the general terminology of *StuffIt* before reading this manual.

System Requirements

I'm not exactly sure what System version a user must have for *WhereSIT* to work. It has been tested on a Mac II and a Mac Plus, both using System version 6.0.2. Theoretically, *WhereSIT* will work on any System which is HFS-compatible. If you have any problems, please contact the author.

Acknowledgements

I would like to thank the members of the Appdev Forum on Compuserve for their help with the development of *WhereSIT*. They contributed several ideas to what problems I faced during the designing of the program, and I hope that I can return the favour (now that I have learned C).

Distribution

WhereSIT is no part of the *StuffIt* or the *StuffIt Utilities* packages by Raymond Lau. The author of *StuffIt* is not responsible for the functioning of *WhereSIT*. If any problems arise in *WhereSIT*'s use, please bother me, not Raymond.

WhereSIT can be distributed freely through BBSs, users' groups, communications networks, exchange forums,

and the like, providing there is no charge except for normal duplication/usage fees. Please keep *WhereSIT*, the manual, and any addendums bundled together in such distribution.

Commercial distribution of *WhereSIT* is strictly prohibited without special licensing from the author.

About the Author

If you have any suggestions, comments, questions, or criticisms, please do not hesitate to contact me. I like receiving mail, be it paper or electric, so please let me know what you think.

The author can be reached at:

Post:
Robert S. T. Gibson
RR#1 Carrying Place,
Ontario, Canada.
KØK 1LØ.

Compuserve: 71261,2236
GEnie: J.GIBSON4

Chapter 2 : **Searching**

Specifying a Key Word

The edit text item (bearing the flashing text entry bar) contains the name you wish to search for. Any file found containing the text entered into this box is labelled as a match and is entered into the list below. A tab, return, or enter character be entered into the search text by holding down the command key and pressing the tab, return, or enter key, respectively.

The search is not cAsE sEnSiTiVe. It is, however, diacritical-sensitive (ë ≠ É).

An empty search key does not abort the search. The empty key causes the first fifty archived files to be added to the list.

In any search, only the first fifty archived files that *WhereSIT* locates and labels as matches are entered into the list. Upon arriving at this count, the search is aborted and a message appears.

The search scans directories and files, and looks for matches in both. Applications, documents, and folders are each displayed with a different icon in the file list. Multiple archived directories (HMF) are transversed, and their contents are also searched. Segmented archives are not searched.

Running the Search

Clicking on the Go icon (with the man-like figure) or hitting enter or return causes the search to execute. Any files listed from a previous scan will be removed from the list before the search begins. Click the stop icon or hit command-period to halt the search.

From left to right, the boxes at the bottom of the window display the following information while searching.:

1: the percentage completed bar is not yet functional. However, since it looks good, I left it there.
2: the current file being examined (this displays all files, even if not an archive).
3: the current folder being scanned.
4: the number of matching archived files or folders found.
5: the number of files scanned.
6: the number of directories scanned.

MultiFinder Use

WhereSIT is MultiFinder-compatible, meaning that even while the search is being executed, you can switch from *WhereSIT* to another application, and the search will continue (although it might be a tad slower). If you have System 6.0 or later, a beep will sound and the apple symbol ( ) will be replaced by a small icon, similar to the Go icon, to signal that the search has been completed. This will notify the user of completion even when *WhereSIT* is not the current application. If too many files are found, an alert will be displayed above all other applications stating so.

File Information

All found files are listed in the list below the search edit text box. You can transverse the list by clicking in the list, dragging through the list, scrolling the list, or by using the arrow keys on your keyboard (providing an item is currently selected).

Selecting a cell entry displays file information in the bottom left hand corner of the window, and displays the path to the file in the bottom right hand corner of the window. The path list can be scrolled by using the scroll bar or by clicking and dragging in the list's region. No cells will be selected in the pathname box. The first entry in the list will always be the *StuffIt* file archive name in which the selected file or folder can be found. If the selected item is a file, the directory level will be displayed in the top right hand corner of the aforementioned file information area. Level 0 signifies the top level, level 1 signifies that the file is directly within a folder in the archive, level 2 signifies that the file is within a folder inside the folder in the top level of the archive, etc. If a folder is selected which displays size information, it is a directory on the top level of the archive. Otherwise, the folder's content level is displayed in the information rectangle.

Narrowing the Search

Although *WhereSIT* does not have the ability to narrow the search for files based on size, date created, etc. (I had it all, and the program turned out to be twice or even three times as big in K), you can narrow the search by selecting the Search Here item in the Preferences menu (either the popup on the window, or that in the menu bar). Using this dialog, you can open the folder you wish to search in. The folder must be opened to be the first one searched. That is, simply selecting the folder of your choice, but not opening it, will cause the directory previous to the one you selected to be searched first. It would eventually be searched, however, since all directories contained within the specified directory are also searched. You can also select another disk to search.

One user-interface point I should make here is that you should remember that hitting enter or return selects the current directory you are in to be your search directory. Unlike the normal file opening dialog box, this does not cause a folder to be opened (which I found quite frustrating at times). I questioned this, but I have stuck with the interface that other file finding utilities offer. Command key equivalents are available for all buttons. Use the first letter of each button as its equivalent.

Use the directory popup menu in the *WhereSIT* window to move closer to the root directory. It acts the same as the menu contained in the standard file dialog.

Moving to the Desktop

The Move to Desktop option, only active when a file or folder is in the current selection, moves the archive containing the selected file to the desktop level. It will appear in the background area behind all windows. All entries in the list which are members of the moved archive are deleted from the list.

When you are finished with the file, just select the Put Away option in the File menu of the Finder after selecting the file, and it will be moved back to its original directory.

Appendix A: **Conclusion**

Legalities

Since I'm not really into trademarks and such, I'll put them here in the appendix.

Any names and/or information that are registered trademarks of any company, corporation or group are hereby respected and acknowledged.

Bug Reports

Please send any and all bug reports to the author at the address given in Chapter 1, section "About the Author". Don't hesitate to write or send E-Mail, please. I would appreciate very much hearing from you, even if what I hear is a bug report. Making a utility is much more satisfying if one gets a response.

Future Plans

The future of *WhereSIT* depends on your response to it. If I get no response at all, I won't be adding much to it. However, if people do give me ideas, report bugs, etc., *WhereSIT* will continue to be supported. Any updates to *StuffIt* will also hopefully be supported by *WhereSIT*.

Suggestions on the following topics are requested:

With System 7.0 coming, which supports small tool applications like DAs, is it worthwhile making *WhereSIT* into a DA? MultiFinder users, remember that MultiFinder treats a desk accessory the same as a separate application. Doing so would require me to learn how to write a DA. To tell you the truth, I'm not exactly sure that it would be worthwhile converting WhereSIT into a DA.

What other search narrowing functions should I implement which are not too cryptic. I have a version which has such search facilities as four key names, three size inequalities, date restrictions, and info restrictions, for both archives and contained files. This turned out to be a bit too much, since each entry is separated by brackets and a logical AND/OR. It's actually powerful, but very complicated. It's also not suitable for an archive searching utility.

Please report your suggestions to the address given above.

Please see the about box for information on *WhereSIT*. I should mention that part of the about box routines were provided by THINK in their release version of *LightspeeedC* 3.0.

The Helvetica font is required for smoother animation of the words. You will get a message if Helvetica is not installed fully, right above my CIS and GEnie IDs. Since it might be a bit confusing looking at my GEnie ID, I'll remention it: J.GIBSON4. Yes, in the about box, its a period, not a star.

Appendix B: **Addendums**

Addendum I : **Bug Fix and Grep Searching**

Special Thanks

I would like to take this chance to thank Ross W. Smith, who helped me nail down one bug that had been plaguing his system (mentioned below). It's great to hear responses and offerings of help from users of my programs.

I also thank Jeff Swart for suggesting pattern (grep) searching to me.

1.0.1 Update

Version 1.0.1 Bug Fix

Version 1.0.1 was a non-release version of *WhereSIT*, featuring a fix of a problem with the Search Here option. On certain systems, it would result in different crashes when selected. Thanks to Ross W. Smith, who notified me of the bug and tested the program on his machine (which was the only one I could find that the worst crashes erupted from), the bug was dispersed.

1.1 Update

Version 1.1 Feature Addition

WhereSIT version 1.1 now includes grep or pattern searching and a few cosmetic fixes and additions.

Choosing Options

Click on the Options box on the top right corner of the editable text box, and the Options dialog will appear. The main dialog and the menus will become dimmed to show that you must close the Options dialog before using the other features.

The Ignore Case button tells *WhereSIT* whether you want case to be acknowledged in the search. That is, "cAsE sEnSiTiVe" is not the same as "case sensitive" if Ignore case is not on, although they are equal if Ignore case is selected. Thus, the case section of "Specifying a Key Word" in Chapter Two in the user's guide is no longer always true.

The Word Match button indicates whether grep should match on words. If word match is on, then the entire search key is treated as if it were surrounded by \< and\> (not any internal word patterns).

The Grep Search item indicates whether the grep search algorithms should be used (mentioned below).

Each button in the dialog has many keyboard equivalents. The first letter of either word in each of the check buttons will select it, and any letter associated with the OK button will select it. The command key is not necessary.

The Options can also be chosen without using the dialog box. When in the main window, simply click on one of

the letters above the editable text item. “I” is associated with Ignore case, “W” is associated with Word match, and “G” is the equivalent of Grep search. The letter is in bold when it is active, and plain when not active. When “W” is disabled (dimmed) it is always inactive. This is ignored on a plain character-comparison (non-grep) search. These items can also be selected by holding down the command key and typing the appropriate letter. Again, if “W” is disabled, it cannot be changed.

The options from your session in *WhereSIT* are saved, so the next time you open *WhereSIT*, you will be reverted back to the settings you had on the last session. For example, if you have grep searching on, then it will be saved and will be on again the next time you run *WhereSIT*.

Introduction to Grep

Grep is a method of pattern searching which derives from the Unix™ system, allowing for sophisticated pattern matching operations.

A pattern is a string of characters which describes a set of strings of characters. The string is matched by a pattern if it is a member of the set described by the pattern. Patterns are composed of sub-patterns, which are, in turn, patterns in themselves.

Pattern Matching

Any character, with exceptions listed below, is a pattern that matches itself.
Example: Pattern `x` matches character `x`.
If the Ignore case option is checked, pattern `x` will also match character `X`.

The character `.` matches any character.

The character `\` followed by any character except `(`, `)`, <, >, or one of the digits `1-9` is a pattern that matches the character following the `\`.
Example: Pattern `\.` matches a `.` and pattern `\\` matches a `\`.

The string of characters *x* surrounded by square brackets (`[` and `]`) gives pattern `[`*x*`]`, which matches any one of the characters in the string *x*. The pattern `[^`*x*`]` matches any character that is not in the string *x*. If a string of characters in the form `a-b` appears in *x*, this represents all the characters from `a` to `b` inclusive. All other characters are taken literally. The only way to include the character `]` in *x* is to make it the first character. Similarly, the only way to include the character `-` in *x* is if it appears either at the beginning or at the end of *x*.
Example: Pattern `[A-Za-z0-9]` matches any alphanumeric character. The pattern `[^!-~]` matches any non-printing ASCII character.
The Ignore case option has no effect between brackets: letters must match exactly.

A pattern `x` followed by a `*` gives pattern `x*` that matches zero or more consecutive occurrences of characters matched by `x`.
Example: The pattern `[a-c]*` matches `a`, matches `acbca`, and matches an empty string.
You'll see below why this is useful.

Any sub-pattern of the form `x*` in a pattern `x*y` matches the largest number of occurrences of whatever `x` matches that still allows a match to `y`. In matching `(())` against the pattern `(.*)`, only the inner pair of parentheses matches the sub-pattern `.*`, so the pattern will match `(())`.

A pattern surrounded by `\(` and `\)` matches whatever the sub-pattern matches.
Example: `\(a[x-z]c\)` matches the same thing as `a[x-z]c`.

A `\`, followed by *n*, where *n* is a digit `1-9`, matches whatever the *n*th `\(` sub-pattern matched. A `*` can be added to a `\`*n* to form a pattern `\`*n*`*` that matches zero or more occurrences of whatever `\`*n* matched.
Example: To find two repeated words, you might use a pattern like this:

`\([A-Za-z0-9][A-Za-z0-9]*\) \1`. This pattern matches a space, any sequence of alphanumeric characters, a space, and the same sequence of alphanumerics. `\1` is not a reapplication of the pattern. Instead, it becomes whatever the `(\)\` pair matched.

A pattern bounded by \< and \> matches whatever the pattern matches, provided that the first and last characters of the string match [A-Za-z0-9_] and that the characters immediately surrounding the matched string don't match [A-Za-z0-9_] (ie. can be matched by [^A-Za-z0-9_]).The pattern matches only if the string begins and ends on a word boundary. If the Word match option is selected, the entire pattern you enter is treated as though it were surrounded by \< and \>.
Example: To find occurrences of repeated words even if they're not surrounded by spaces, you would use the pattern \(\<[A-Za-z][A-Za-z]*\>\)[^A-Za-z]*\1.

A pattern x preceded by a ^ gives pattern ^x. If ^x is not preceded by any other pattern, it matches whatever x matches as long as the first character that x matches occurs at the beginning of a line.

A pattern x followed by a $ gives pattern x$. If the pattern x$ is not followed by any other pattern, it matches whatever x matches as long as the last character that x matches occurs at the end of a line. If the pattern x$ is followed by any other pattern, then the $ is taken literally.

Some Examples:

Pattern	Status	Text	Ignore case
X	matches	...X...	
	doesn't match	...x...	on
	matches	...x...	off
Sit	matches	...Sit...	
	doesn't match	...sit...	on
	matches	...sit...	off
w.yz	matches	...wxyz...	
	matches	...wXyz...	
w\.yz	matches	...w.yz...	
	doesn't match	...wxyz...	
w\\yz	matches	...w\yz...	
[xyz]	matches	...xabc...	
	matches	...a**x**c...	
	doesn't match	...**xy**...	
a[xyz]	matches	...ay...	
	doesn't match	...aY...	
[xyz][pqr]	matches	...xq...	
	doesn't match	...pq...	
[a-c]	matches	...ac...	
	and matches	...x**c**...	
[ax-z]q	matches	...aq...	

	and matches	...xq...
[-a-c]	matches	...-a...
	and matches	...xa...

Pattern	Status	Text	Ignore case
`[a-c]*`	matches	...a	
	and matches	...acbca	
	and matches	nothing	
`A[a-z]*`	matches	...A...	
	and matches	...abcb...	
	and matches	...abc...	on
	doesn't match	...abc...	off
`(.)*`	matches	...(aaa)...	
	and matches	...()...	
`\(abc\)`	matches	...abc	
`\(ab(\)`	matches	...ab(	
`\(abc(\)\1`	matches	...abcabc...	
`\(a.c\)\1`	matches	...axcaxc...	
	does not match	...axcayc...	
	does not match	...axcaXc...	
`\<ab*\>`	matches	...+ab+...	
	does not match	...+abc+...	
`^ab*`	matches	ab...	
	does not match	xab...	
`ab^ab*`	matches	...ab^ab...	
`xy$`	matches	...xy	
	does not match	...xya	
`xy$xy`	matches	...xy$xy...	
`^ab$`	matches	ab	
	does not match	ab...	

Using Grep

When executing a grep search, remember that if your pattern makes a match anywhere in the string, the search will return a match and the file will be added to the list.

You should keep the Note Pad or another such utility close at hand so you can keep complicated search patterns and have easy and quick access to them. *WhereSIT* supports the usual Cut, Copy, and Paste commands, so you can easily transfer your patterns into another file. An easy way to select the entire search text is to press the tab key when the main window is active.

Addendum II : **Searching and Exporting**

Special Thanks

I would like to thank those who left messages to me concerning *WhereSIT*. If you have any comments, criticisms, or suggestions, please send me a message. I like receiving mail, be it paper or electric, so please let me know what you think.

I can be reached at:

Post:
Robert S. T. Gibson
RR#1 Carrying Place,
Ontario, Canada.
KØK 1LØ.

Compuserve: 71261,2236
GEnie: J.GIBSON4

1.2 Update

Version 1.2 Feature Additions (no bugs reported, no bugs fixed)

WhereSIT 1.2 now features normal file searching capabilities, allowing you to search your disks for uncompressed files as well as Stuffed. You may also export your list of files to a text file, suitable for use with any word processors or spreadsheets. Again, a few cosmetic and usage changes were made.

File Searching

WhereSIT now allows you to specify to search for a Stuffed file, an uncompressed file, or both. The Stuffed files and Normal files button (controlled by command-S and command-N, respectively) control what files you wish to search for. Like the options dialogs, your preferences are saved from one *WhereSIT* execution to the next.

Uncompressed folders and files each display different information in the information box. The Move to desktop preference is functional with all files and folders in the list. Files within folders moved to the desktop can be moved to the desktop themselves as well.

Export Options

Before attempting the catalog export, you should check your exporting options. In it, you can specify what types of files to include in your catalog. Most of the file types are self-explanatory. Stuffed sub-folders are folders which lie inside another directory. You can tell if a folder is a sub-folder by looking at its information. In the information box, only its level is displayed. In the list, the sub-folder contains a "--" under the Size category. When looking at the Size category, don't confuse the sub-folder with the uncompressed folder. Stuffed sub-folders have an archive name under the Archive heading, while uncompressed files and folders contain the "--" indicator.

The Format text box contains the information you wish to include in your catalog. Different letters stand for the categories. Case is ignored except where noted.

Indicator	Displays
K	Creation date

k	Creation time
M	Date of last modification
m	Time of last modification
B	Backup date (usually 1/1/4)**

b	Time of last backup (usually 12:00 AM)**
P	Pathname to file, folder, or archive
A	Archive name*
N	File name
D	Uncompressed data length††
d	Compressed data length††
U	Data compression method*††
R	Uncompressed resource length††
r	Compressed resource length*††
Y	Resource compression method*††
C	File creator†
T	File type†
,	Separator indicator (discussed below)
\	A literal indicator. Any character following a \ is taken literally, be the character a predefined indicator or not. Note that \\ gives a \ in the output.

* Undefined if uncompressed
** Undefined if Stuffed
† Undefined if a folder
†† Undefined if a folder other than a first-level Stuffed folder

Undefined displays are considered empty strings when output to the catalog.

Any literal characters are displayed exactly as shown in the Format box for each record. A limit of 255 characters is placed on the size of the Format indicators (but not necessarily each output record, of course), and a 32K limit has been placed on the catalog size.

Use the Separator box to determine what characters should be displayed at each separator indicator (a comma (“,”)). For a tab-delimited file, select the text in the box and replace it with a command-tab. The separator indicator will be replaced by the Separator text.

Set the file type and creator using their respective edit boxes. Remember that types and creators are restricted to four characters.

Standard Cut, Copy, Paste, and Clear commands are functional through their command key equivalents in the Export Options dialog. It might be wise to make your Format options in another program (running in a larger display rectangle) to make editing easier.

For ResEdit hackers: The options are stored in an EXPT resource. To prepare a library of different formats, just copy the resource from *WhereSIT* after making in *WhereSIT* it and quitting from the application.

Export Catalog

Once the options are properly set, simply select the Export catalog menu item and select the file name and destination directory. If a file is already present with the specified name, it will be replaced. A new file with type and creator specified in the Export options will be created and the catalog will be made. The export action usually does not take long. A percentage bar will show you the amount of work done and remaining to be completed.

General Additions

I added a few things to *WhereSIT* which might make it easier to use. Most of them are small features that are apparent when you use it that I won't mention here. However, a few keyboard equivalents need some descriptions.

Pressing command-backspace when an item is selected in the list causes that item to be deleted.

Pressing the page up or command-up arrow moves the list up one screen. Similarly, page down and command-down arrow cause the list to scroll down one screen.

The home key selects the topmost entry in the list and scrolls to the selected item. Similarly the end key moves the list and selection to the last entry.

The help key displays the About window, with color display (if your Mac can display in color).

The forward delete key deletes the character after the selection point. If there is no character after the cursor, the forward delete key acts as the normal delete or backspace key.