

WINDOW

SCREEN EDITOR USER GUIDE

by John Walker

Marinchip Systems

Mill Valley, CA 94941

(C) Copyright 1980 Marinchip Systems

All Rights Reserved

This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

Marinchip Systems makes no warranty, either express or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding these materials and makes such materials available solely on an "as-is" basis.

In no event shall Marinchip Systems be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of purchase or use of these materials. The sole and exclusive liability to Marinchip Systems, regardless of the form of action, shall not exceed the purchase price of the materials described herein.

For condition of use and permission to use these materials for publication in other than the English language, contact Marinchip Systems.

Marinchip Systems reserves the right to revise and improve its products as it sees fit. This publication describes the state of this product as of the time of its publication, and may not reflect the product at all times in the future.

This manual was prepared and published in December of 1980, and is based on Release 1.0 of Window.

Window User Guide - Table of Contents

1.	Introduction	1
1.1.	Learning to use Window	1
1.2.	Notation and conventions	1
1.3.	Configuration requirements	2
2.	Calling Window	3
3.	Screen Mode	4
3.1.	Graphic characters	4
3.2.	Control characters	4
3.2.1.	Editing within a line	4
3.2.1.1.	Cursor Right (Control L)	4
3.2.1.2.	Cursor Left (Control H, Backspace)	4
3.2.1.3.	Tab (Control I, Tab)	5
3.2.1.4.	Beginning of Line (Control U)	5
3.2.1.5.	End of Line (Control N)	5
3.2.1.6.	Next Word (Control A)	5
3.2.1.7.	Previous Word (Control W)	5
3.2.1.8.	Erase Character (Rubout, Del)	5
3.2.1.9.	Delete Character (Control D)	5
3.2.1.10.	Delete Word (Control V)	6
3.2.1.11.	Delete to End of Line (Control Y)	6
3.2.1.12.	Erase Line (Control Del, Control Shift O)	6
3.2.1.13.	Insert Character (Control F)	6
3.2.2.	Moving between lines	6
3.2.2.1.	Cursor Down (Control J, Line Feed)	6
3.2.2.2.	Carriage Return (Return, CR, Control M)	6
3.2.2.3.	Cursor Up (Control K)	7
3.2.2.4.	Home (Control ^, Control Shift N)	7
3.2.2.5.	File End (Control E)	7
3.2.2.6.	Page Down (Control T)	7
3.2.2.7.	Page Up (Control R)	7
3.2.2.8.	Center Cursor (Control S)	7
3.2.2.9.	Delete Line (Control Z)	7
3.2.2.10.	Concatenate Lines (Control O)	8
3.2.2.11.	Insert/Split Line (Control B)	8
3.2.2.12.	Next Match (Control X)	8
3.2.3.	Mode commands	8
3.2.3.1.	Escape (Escape, Alt Mode, Control [)	8
3.2.3.2.	Character Insert Mode (Control Q)	8
3.2.3.3.	Entry Mode (Control \, Control Shift L)	9
3.2.4.	Marking blocks of lines	9
3.2.4.1.	Define Block (Control P)	9
4.	Command Mode	11
4.1.1.	File commands	11
4.1.1.1.	FINISH (F) - Update file and exit	11
4.1.1.2.	SAVE - Update file	11
4.1.1.3.	QUIT (Q) - Discard updates	11
4.1.1.4.	GO (G) - Exit and process output	12
4.1.1.5.	ERASE - Delete file	12
4.1.2.	Display mode commands	12
4.1.2.1.	TAB (T) - Set tab stops	12
4.1.2.2.	MARGIN (M) - Set display left margin	12
4.1.3.	Context search and replacement commands	12
4.1.3.1.	SEARCH (S) - Search for text in file	14

Window User Guide - Table of Contents

4.1.3.2.	NEXT	- Search for next occurrence of text	14
4.1.3.3.	REPLACE (R)	- Replace text	15
4.1.3.4.	GLOBAL	- Global replacement	15
4.1.4.	Block commands		15
4.1.4.1.	DELETE	- Delete marked block	15
4.1.4.2.	COPY	- Copy marked block	16
4.1.4.3.	MOVE	- Move marked block	16
4.1.4.4.	CLEAR	- Clear marked block	16
4.1.5.	Formatting commands		17
4.1.5.1.	CENTER	- Center text on line	17
4.1.5.2.	FORMAT	- Reformat paragraph	17
4.1.5.3.	WIDTH	- Set line width for formatting	18
4.1.6.	Cursor positioning commands		18
4.1.6.1.	LINE	- Move cursor by lines	18
4.1.6.2.	PAGE (P)	- Move cursor by pages	18
4.1.6.3.	PUSH	- Save current position	19
4.1.6.4.	POP	- Return to previous position	19
4.1.7.	Mode switch commands		19
4.1.7.1.	INDENT	- Automatic indentation	19
4.1.7.2.	SBLK	- Limit searches to marked block	20
4.1.7.3.	SEARCHFOLD	- Match letters regardless of case	20
4.1.7.4.	TEXT	- Allow next/last word between lines	20
4.1.7.5.	WRAP	- Move words between lines	21
4.1.8.	Miscellaneous commands		21
4.1.8.1.	HELP	- Print command summary	21
4.1.8.2.	BLANKS	- Insert blank lines	21
4.1.8.3.	EDIT (E)	- Return to Screen Mode	21
5.	The Window Configuration File		23
5.1.	Mode statements		23
5.2.	GOCMD	- Define postprocessor	24
5.3.	PAGE1	- Specify first paging file	24
5.4.	PAGE2	- Specify second paging file	24
5.5.	IF and ENDF	- Conditional configuration	25
5.6.	Sample configuration file		25
6.	Editing large files		27
6.1.	Default paging files		27
6.2.	User-defined paging files		27
6.3.	Paging error conditions		28
7.	Generating and Configuring Window		29
7.1.	The Window release discs		29
7.2.	Selecting an input driver		29
7.3.	Selecting an output driver		30
7.4.	Linking Window		31
7.5.	Installing Window in your system		31
7.6.	For more information on drivers		32
8.	Terminal reference guides		33
8.1.	Standard terminal input		34
8.2.	Hazeltine terminal input		35
8.3.	IBM 3101 terminal input		36

Window User Guide

1. Introduction

Window is a screen-oriented text editor developed by Marinchip Systems for use on its M9900 computer systems. By taking advantage of the features of modern high-speed video display terminals, Window is able to provide a means of online editing that is both powerful and easy to learn.

The basic concept of Window is to turn the user's terminal into a window into the file being edited (hence the name). The window may be moved up and down or left and right through the file. A CURSOR, or data pointer, can be moved anywhere within the window, and "drass" the window after it as you attempt to move it outside the window. Data in the file may be modified simply by overtyping characters with the characters desired to replace them.

Moving within the window, moving the window through the file, inserting and deleting characters and lines, and many other functions are performed by typing CONTROL CHARACTERS on the keyboard. While text characters replace characters in the window, control characters act as commands to Window and cause it to take a specified action. Window cannot be used to create or edit files which contain control characters.

For editing functions which are too complex to express in a single keystroke, Window provides a COMMAND MODE. This mode may be entered by pressing a key, and may be exited also by a single key. While in Command Mode, the display of the window into the file is replaced by a special Command Mode display, and Window will accept and process commands which perform various functions such as searching the file for specified text, performing global or local changes based on context, copying, moving, or deleting blocks of lines, and so forth. Upon exit from Command Mode (unless a command was deliberately used to change things), the terminal returns to the display of the window into the file, right where you left it.

1.1. Learning to use Window

This manual has been written to be both a complete definition of how Window operates and a ready reference guide organized so that commands may be looked up conveniently while using Window. The organization of the manual is not designed to take beginners through the package. This is not a serious problem, as the best way to become familiar with Window is to use it. At first, you will have to refer to this guide continuously, but after an hour or two, most of the commands you'll use will become "second nature". Window is easy to learn and use because it adheres to the "rule of visual fidelity", which is a high-falutin' way of saying "what you see is what you get".

1.2. Notation and conventions

Because terminal keyboards differ so much, it is not possible to give a single description of Window commands for all terminals. For example, on one terminal, a specific character is generated by pressing Control and "\" simultaneously, while on another terminal, you must press Control, Shift, and "L" to generate the same character.

Window User Guide

In this manual, when we list the control characters in section titles we will first identify the character by its function within Window (for example, "Cursor Right"), then give the most common ways of generating the standard code for this command.

Many terminals have special keys dedicated to functions such as moving the cursor, going to the top, inserting and deleting characters and lines, etc. If these keys operate properly in full-duplex mode and a proper driver has been written for Window for that terminal, the special keys on the terminal may be used instead of Window's normal control characters. Wherever possible, we design such special drivers so that either the special key or the control character described in this manual may be used. However, due to the great effort some terminal manufacturers expend to invent new ways to be incompatible, there are exceptions to this rule.

For all special terminal drivers we supply, there is a reference sheet at the back of this manual which gives the Window commands and the key assigned to that command on the terminal. These sheets are the final arbiter of how Window behaves when configured for a specific terminal.

1.3. Configuration requirements

Window operates under all Marinchip operating systems and will run on any M9900 configuration with 48K bytes of memory or more. Adding memory to the system increases the efficiency of Window, but is not necessary to edit a large file, as Window incorporates a disc-based virtual storage system which permits even the largest files to be edited on a small configuration.

Because of the enormous variety of video terminals available today, and because virtually no two terminals are alike in keyboard layout and command sequences, Window has been designed to permit the user to configure it for the specific terminal it will be used with. Marinchip Systems provides drivers for several popular and representative terminals; these may be used as models for developing your own. Window has been designed to work on any terminal supporting the following minimum set of commands:

- Clear screen
- Address cursor
- Scroll up (line feed)

If the terminal has additional commands such as clear to end of line, insert line, and delete line, Window will take advantage of them to minimise the amount of data it must rewrite to effect changes on the screen. However, all operations in this manual will work on any terminal with the minimum set of commands needed to run Window.

Window User Guide

2. Calling Window

To use Window to edit a file, simply type the command:

```
WINDOW <file name>  
or WINDOW <output file>=<input file>
```

while at operating system command level (assuming that Window has been placed on the system disc (for Disc Executive) or in the system library (for NOS)).

If Window is called with a single file name (as in the first example above), it will read the file in, let you edit it, then write the modified data back to the same file name. If the named file does not exist, and Window is being used under NOS, it will be automatically created, and Window will start with a null file so that you may enter data.

If Window is called with two file names, the first file name is used as the output file and the second file name is used as the input file. If no output file is given, Window will operate in "read only mode". That is, it will read in the input file, but not write any output when you leave Window. If no input file is specified, Window assumes that it is creating the output file from scratch, and starts with an empty file. When omitting the input file or output file, the equal sign must still be supplied.

When Window is called, it first prints its sign-on message including the release version number and, assuming that a valid input file has been specified, prints a message confirming that it is loading that file into its internal workspace. Just to let you know something is happening, Window will print a plus sign, "+", for every 100 lines it reads in from the file being loaded.

After the file has been loaded into Window's internal memory, the screen will clear and the first page of the file will appear on the terminal. The cursor will be positioned at the first character of the first line of the file. You are now in SCREEN MODE, and may edit the file by using the commands described in the next chapter.

Window User Guide

3. Screen Mode

While in Screen Mode the terminal display is a window into the file being edited. As described below, graphic (text) characters simply overwrite data in the window and hence in the file. The various control characters described serve to move the cursor within the window, move the window within the file, delete and insert characters and lines, and so on.

If the window extends beyond the end of the file, lines on the screen corresponding to lines past the end of the file will be printed as all underlines, "_". If the file is extended by typing data past the end of the file, the underlines will vanish and be replaced by the data typed on the new line.

3.1. Graphic characters

When in Screen Mode, all of the graphic characters (letters, numbers, punctuation) simply overwrite characters already in the file. Blanks are like any other character; if you type a blank, it replaces any character previously in the file. It is important to understand the difference between typing a blank, which will replace the character already in the file, and simply moving the cursor (with the commands described below), which leaves data in the file unchanged.

When you type a graphic character, it is inserted at the current cursor position and the cursor moves right one character. If the cursor would run off the right end of the screen by moving right one character, the window into the file will be moved right by one quarter of the screen width, so that the cursor will remain on screen. Note that it is possible to edit lines much wider than the width of the screen by moving the window left and right.

3.2. Control characters

The following sections describe the actions of the various control characters recognised by Window.

3.2.1. Editing within a line

This group of control characters performs functions within a single line of the file. These characters are described below.

3.2.1.1. Cursor Right (Control L)

The Cursor Right key moves the cursor one position to the right within the line. Characters within the line are not affected by this command. If the cursor is already at the right edge of the screen, this key will cause the window to be moved to the right one quarter of the screen width so that the cursor will remain on the screen.

3.2.1.2. Cursor Left (Control H, Backspace)

The Cursor Left key moves the cursor one character to the left, leaving characters it moves over unchanged. If the cursor is already at the first character of a line, the Cursor Left key will be ignored.

Window User Guide

If the cursor is at the leftmost character of the screen, but not at the start of the line (because the window is shifted to the right in the file), the window will shift back so that the cursor will remain on the screen.

3.2.1.3. Tab (Control I, Tab)

The Tab key causes the cursor to move right to the next tab stop. Tab stops are set with the TAB command in Command Mode. If the cursor is to the right of the last tab stop, the Tab key will move the cursor one character to the right. Moving the cursor with the Tab key does not affect characters it moves over and is equivalent to pressing the Cursor Right key sufficient times to advance the cursor to the next tab stop.

3.2.1.4. Beginning of Line (Control U)

The Beginning of Line key resets the cursor to the first character of the current line. If the window has been moved right through the file, it will be moved back to the left so that column 1 of lines in the file will appear at the left of the screen.

3.2.1.5. End of Line (Control N)

The End of Line key moves the cursor to the end of the data in a line. This is normally just after the last nonblank character in the line (although it is possible to put spaces at the end of a line by going there and typing them). The window will be moved if necessary to keep the cursor on the screen.

3.2.1.6. Next Word (Control A)

The Next Word key moves the cursor to the start of the next word. The window will be moved to the right if required to keep the cursor on the screen. If the end of the line is reached, the cursor will stop past the end of data in the line (except if Window has been placed in TEXT mode, which will cause the Next Word key to move to the first word on the next line).

3.2.1.7. Previous Word (Control W)

The Previous Word key moves the cursor to the start of the previous word on the line. The window will be moved to the left if necessary to keep the cursor on the screen. If the start of the line is reached, the cursor will normally stop there (except if Window has been placed in TEXT mode, in which case the cursor will move up to the previous line).

3.2.1.8. Erase Character (Rubout, Del)

The Erase Character key causes the cursor to back up one character and replace the character it backed up over with a blank. This key is normally used to correct typing errors when entering data. If Window is in Character Insert Mode, the Erase Character key will cause the character to be totally deleted rather than just blanked out.

3.2.1.9. Delete Character (Control D)

Window User Guide

The Delete Character key causes the character at the current cursor position to be deleted from the line. All characters to the right of the character deleted move left one position.

3.2.1.10. Delete Word (Control V)

The Delete Word key deletes all characters starting at the current cursor position until the start of the next word. If the cursor is positioned at the start of a word, that word and any punctuation and blanks following it will be deleted. If the cursor is in the middle of a word, the rest of that word and all characters up to the start of the next word will be deleted. The rest of the line will shift left to take up the deleted space.

3.2.1.11. Delete to End of Line (Control Y)

The Delete to End of Line key deletes all characters starting with the character at the cursor position to the end of the line.

3.2.1.12. Erase Line (Control Del, Control Shift O)

The Erase Line key deletes all characters in the current line. Note the distinction between deleting all characters in a line (which leaves it in the file as a blank line) and deleting the line itself.

3.2.1.13. Insert Character (Control F)

The Insert Character key inserts a blank at the current cursor position and moves the rest of the line to the right. To insert several characters into a line, you must type the Insert Character command once for each character to be inserted so as to make room for all the new characters (or you can use Character Insert Mode, which is independent of this command).

3.2.2. Moving between lines

The following commands move the cursor between lines and perform editing functions that affect lines rather than characters within lines.

3.2.2.1. Cursor Down (Control J, Line Feed)

The Cursor Down key moves the cursor to the next line in the file. The cursor remains at the same character position within the line. If the cursor reaches the end of the screen, lines on the screen will roll up to make room for the next line. If the cursor is at the end of the file, Window will not allow it to be moved past the end with the Cursor Down key.

3.2.2.2. Carriage Return (Return, CR, Control M)

The Carriage Return key moves the cursor to the start of the next line in the file, moving the window left and/or scrolling up the screen if necessary. Using the Carriage Return key while on the last line of the file will place the cursor past the end of the file. If a character is typed while the cursor is past the end of the file, a new line will be appended to the file, and hence the underline display

Window User Guide

will be replaced by the character entered. Thus, new lines may be added at the end of the file simply by typing them in, using Carriage Return to go to each new line. If Carriage Return is entered while past the end of the file, a blank line will be appended to the file and the cursor will move down to the next line, which is now past the end of the file. If Window has been placed in Entry Mode, the Carriage Return key will cause a new line to be placed in the file. See the description of Entry Mode below for details.

3.2.2.3. Cursor Up (Control K)

The Cursor Up key moves the cursor to the previous line in the file. If the cursor is already on the first line in the file, the Cursor Up key will be ignored. If the Cursor Up command is used when the cursor is at the top of the screen (but not the top of the file), the action taken depends on the type of terminal being used. If the terminal supports scrolling down, all lines on the screen will move down one, and the previous line in the file will appear at the top of the screen. On more primitive terminals, the window will move up half the screen height, and the cursor will be displayed on the previous line, but now in the middle of the screen. In either case, the effect of the key is the same; only the representation on the display differs because of varying terminal features.

3.2.2.4. Home (Control ^, Control Shift N)

The Home key positions the cursor at the first character of the first line in the file, rewriting the screen if that position is not in the current window.

3.2.2.5. File End (Control E)

The File End key positions the cursor at the start of the last line in the file. The screen is written so that the last quarter of the screen is past the end of the file (and hence displayed as underlines) so that data may be inserted into that space more conveniently.

3.2.2.6. Page Down (Control T)

The Page Down key moves the cursor forward in the file a number of lines equal to the height of the terminal screen. This command is useful for rapidly scanning through a file.

3.2.2.7. Page Up (Control R)

The Page Up key moves the cursor up in the file a number of lines equal to the height of the terminal screen.

3.2.2.8. Center Cursor (Control S)

The Center Cursor key rewrites the display so that the line on which the cursor is positioned will appear in the center of the screen. This command is especially useful when you want to see the context surrounding a specific line.

3.2.2.9. Delete Line (Control Z)

Window User Guide

The Delete Line key deletes the line the cursor is presently on from the file. All lines below the deleted line move up one line. The cursor will be at the same character position as before, but on the line following the line just deleted.

3.2.2.10. Concatenate Lines (Control O)

The Concatenate Lines key causes the line below the line where the cursor is positioned to be placed at the end of the line the cursor is on. The line below the cursor is then deleted from the file and lines below it move up one. Note that the character position of the cursor is irrelevant to the Concatenate Lines command, and is not changed by the command.

3.2.2.11. Insert/Split Line (Control B)

The Insert/Split Line key inserts a new line into the file at the current cursor position. All characters to the left of the cursor remain on the current line with the cursor at the end. The character at the cursor position and all characters to its right form a new line which is inserted after the line where the cursor is positioned. All subsequent lines move down one line. Note that if the cursor is positioned at the start of a line, this will cause a blank line to be inserted at the current cursor position and the line the cursor was on to move down one line in the file. This command is most often used at the start of the line to make room for a new line to be inserted in the middle of a file. When inserting several lines, it is generally more convenient to use Entry Mode (described below).

3.2.2.12. Next Match (Control X)

The Next Match key works in conjunction with the SEARCH command which is described in the chapter on Command Mode below. When the Next Match key is pressed, the cursor will advance to the next occurrence of the string last used in a SEARCH or NEXT command, starting at the current cursor position. If there are no occurrences of the string between the current cursor position and the end of the file, Window will enter Command Mode.

3.2.3. Mode commands

The following control characters affect operating modes of Window and hence modify the action of other commands.

3.2.3.1. Escape (Escape, Alt Mode, Control [)

The Escape command causes Window to leave Screen Mode and enter Command Mode. Command Mode is described in the next chapter.

3.2.3.2. Character Insert Mode (Control Q)

The Character Insert Mode key acts as a push/push switch, with each depression reversing the state of the Character Insert Mode switch. When Window is in Character Insert Mode, the letter "Q" will be displayed in the upper right corner of the screen. Character Insert Mode allows characters to be entered in the middle of a line without having to press the Insert Character key for each one. When

Window User Guide

Character Insert Mode is on and a character is typed, all characters to the right of the cursor will move over to make room for the new character. In addition, the Erase Character key will back up the cursor and delete the character it backs up over, permitting easy correction of typing errors when entering data in this mode. Thus, to insert in the middle of a line, position the cursor where the new characters should go, enter Character Insert Mode, type the new characters, then exit Character Insert Mode by pressing the key again.

3.2.3.3. Entry Mode (Control \, Control Shift L)

The Entry Mode key acts as a push/push switch controlling Line Entry Mode. When Window is in Line Entry Mode, the letter "I" will be displayed in the upper right corner of the screen. In Line Entry Mode, pressing the Carriage Return key will cause a new blank line to be inserted after the line the cursor is on, and the cursor to be moved to the start of the new line. Line Entry Mode is normally used when adding several lines in the middle of a file. To add lines to the middle of a file, go to the line after which the new lines are to be inserted and enter Line Entry Mode. Then press Carriage Return, which will create the first new line. Enter the data for that line, then press Carriage Return, which will create the next line and move down onto it. At the end of the last line, press the Entry Mode key again, turning off Line Entry Mode.

3.2.4. Marking blocks of lines

One control character is used to define a block of lines to be operated on by various commands in Command Mode. The action of this character is as follows.

3.2.4.1. Define Block (Control P)

To mark a block of lines, move the cursor to one end of the desired block and press the Define Block key. Then move the cursor to the other end and press the Define Block key again. The block will then be defined as the two lines on which the cursor was positioned when Define Block was pressed, and any lines between them. To define a single line as a block, simply press Define Block twice on the same line.

When the first line is marked, the screen will be rewritten with all lines displayed starting in column 2 of the terminal screen. The line marked with the Define Block key will be indicated by being preceded by a plus sign. When the other end of the block has been defined, all lines in the block will be identified by being displayed preceded by a plus sign. While a block of lines remains defined, the width of the terminal window into the file is reduced by one character so that the leftmost character may be used to identify the lines within the block. The window may be moved left and right as always; it is just one character narrower than usual.

The limits of the block may be altered by using the Define Block key. If the Define Block key is pressed while the cursor is above the top of the block, the block is extended upward to the new cursor position. If the cursor is at or below the top of the block, the line where the cursor resides becomes the new bottom of the block.

Window User Guide

The blocks defined with the Define Block key are blocks of lines. The position of the cursor within a line when the Define Block key is pressed is irrelevant.

Once a block has been defined, you are free to edit within it, including adding and deleting lines within the block. Window will still consider all lines between the marked lines to be part of the block. If you delete the top line of the block, the next line will become the top. If you delete the bottom line of the block, the previous line will become the bottom. If the block is only a single line and you delete it, no block will be marked following the deletion.

4. Command Mode

Command Mode is entered by pressing the Escape key while in Screen Mode. This key causes the display of the window into the file to vanish and be replaced with the Command Mode display. Window will print a heading which looks like the following:

```
Window 2.7 Command mode. Editing ZORCH.BAS Length 117 lines.  
Cursor at line 87 character 19 Screen top at line 81 character 1  
Modes: Searchfold Indent  
Line range defined. Top: 61 Bottom: 73  
Tabs: 9 12 15 18 21
```

This heading is a brief summary of the state of Window. It gives the release version of Window, the name of the file you're editing (just in case you forgot) and the current length of that file. The second line tells you the cursor position within the file, and where the top of the window is. The next three lines appear only if applicable. If any of Window's mode switches have been turned on, they will be listed. If the Define Block key has been used to mark a block of lines, the block will be identified, and any tab stops set will be listed.

Following the heading, Window prints a right corner bracket, ">", as its command prompt and awaits a command. If you simply press Carriage Return, Window will return to Screen Mode, right where you left it. Most often, of course, you will enter one of the commands described below. Window permits the most commonly used commands to be abbreviated to a single letter. The sections describing those commands list the abbreviations in parentheses after the command name.

4.1.1. File commands

The following commands affect the disposition of the file being edited, or other files.

4.1.1.1. FINISH (F) - Update file and exit

FINISH

The FINISH command causes the updated data to be written back to the output file name given on the original call to Window. After the data are written back, Window exits to system command mode.

4.1.1.2. SAVE - Update file

SAVE

The SAVE command writes the current state of the file back out to the output file named on the call to Window, then returns you to Screen Mode. This command is primarily used periodically during a lengthy editing session to protect against loss of all the work done on the file in case of power failure or other catastrophe.

4.1.1.3. QUIT (Q) - Discard updates

Window User Guide

QUIT

The QUIT command causes Window to exit without writing the updated file back out. This command is used to leave Window when you were "Just looking" at a file and don't want to have to wait for it to be written back out, or when you have made some horrible mistake while editing and find it better to start over than to try to undo the error.

4.1.1.4. GO (G) - Exit and process output

GO

The GO command writes out the updated file, then invokes the program normally used to process the output. The processor is selected based on the file type (the name following a period in the file name) which is looked up in the Window configuration file. See the description of the Window configuration file for information on how to set up the GO command. If no configuration file is present, the GO command is equivalent to the FINISH command.

4.1.1.5. ERASE - Delete file

ERASE <file>

The ERASE command deletes the named file. This is simply a way of deleting a file without having to leave Window and come back in.

4.1.2. Display mode commands

The following commands set modes affecting the operation of Screen Mode. They are distinct from other modes, to be discussed later, which affect the operation of various Window commands.

4.1.2.1. TAB (T) - Set tab stops

TAB <number>,<number>,...

The TAB command sets tab stops at the specified columns. The leftmost column is column one. Refer to the description of the Tab key in the Screen Mode chapter for information on how tab stops are used in Screen Mode. Window allows up to 10 tab stops to be defined. Each TAB command replaces the previous stops with the ones given on the new command. A null TAB command turns off all the tab stops, and makes the Tab key in Screen Mode work exactly like the Cursor Right key.

4.1.2.2. MARGIN (M) - Set display left margin

MARGIN <number>

The MARGIN command sets the display in Screen Mode to begin with the specified column. This command is most useful when editing lines wider than the screen, when Window's automatic "dragging" of the window as the cursor is moved is not optimal for the text being edited.

4.1.3. Context search and replacement commands

Window User Guide

The following commands search through the file being edited, or a portion of it, and optionally permit replacement of sequences of characters located with other text supplied by the user.

Text searched for with these commands may be given either explicitly or generically through the use of "wild card" characters. All characters other than those listed below must match exactly. The following characters permit matching groups of characters:

- # Matches any digit (0 - 9).
- ! Matches any letter (upper or lower case).
- ? Matches any character.

If a character is preceded by a backslash, "\", it must be matched exactly. This is used to permit the wild card characters to be searched for. For example, to search for the text:

A 10# cockroach? Impossible!

you would tell Window to search for:

A 10\# cockroach\? Impossible\!

Note that the backslash may be used to force itself when you wish to search for a backslash in the file.

All of the following commands work in basically similar ways, and all respond to the setting of two mode switches, SBLK and SEARCHFOLD. We have not yet discussed mode switches or how to turn them on and off, but all you need to understand about them at this time is that they exist and that they are initially off.

If the SBLK mode is off (Window's normal state), the search and replace commands will work on the entire file. If the SBLK mode has been turned on, the search and replace commands will operate only within the block of text marked with the Define Block key. If SBLK is on but no block of text has been marked, the search and replace commands will always fail to locate the object of their search, since there is nothing for them to search.

The SEARCHFOLD mode controls how upper and lower case letters are handled by the search and replace commands. If SEARCHFOLD is off (Window's normal state), and you are searching for the string:

Turboencabulator

you must specify the word exactly, with an upper case "T", or the command will fail to locate it. If SEARCHFOLD is on, letters will be matched regardless of case, and a search command which specified:

TurboEncabulator

as the object of the search would find all of the followings:

TurboEncabulator

Window User Guide

turboencabulator
TURBOENCABULATOR
TURBoEncaBULator

SEARCHFOLD affects only letters; all other characters must still match exactly.

The SEARCHFOLD mode not only affects how strings are found by the replacement commands, but how the replacement is done. Let us assume that SEARCHFOLD is on, and that we have told WINDOW to replace the word "white" with "purple", and that the file we are editing contains the BASIC statement:

```
WHITE$="There can be no whitewash in the White House."
```

Because SEARCHFOLD is true, all three occurrences of the word "white" will be located as candidates for replacement. To attempt to preserve the context of the word, Window applies the following rules to the word being substituted:

1. If all letters in the old string are capitalised, all letters in the replacement will be capitalised.
2. If the first letter in the old string is capitalised, the first letter in the replacement will be capitalised.
3. Otherwise, the replacement will be inserted unchanged.

These rules will result in our example line being changed to:

```
PURPLE$="There can be no purplewash in the Purple House."
```

which preserves the contextual usage of each occurrence, regardless of whatever damage may be done to the subtle meaning.

4.1.3.1. SEARCH (S) - Search for text in file

```
SEARCH <text>
```

Starting at the top of the file (or the top of the marked block if SBLK mode is on), the file will be searched for an occurrence of the given <text>. If no <text> is specified, the <text> used on the last SEARCH or NEXT command will be used. If the <text> is found, Window will switch to Screen Mode with the cursor positioned over the first character of the located text. If the <text> is not found, the message:

```
String not found in file.
```

will be printed, and Window will remain in Command Mode. Note that once the first occurrence of the <text> has been found with the SEARCH command, subsequent occurrences can be found by using the Next Match key in Screen Mode.

4.1.3.2. NEXT - Search for next occurrence of text

```
NEXT <text>
```

The NEXT command is identical to the SEARCH command except that the

Window User Guide

search starts at the current cursor position rather than at the top of the file (or marked block if SBLK is set). If SBLK mode is on and the cursor is above the top of the marked block, NEXT will begin the search at the top of the marked block. If SBLK mode is on and the cursor is below the marked block, NEXT will give the message:

Strings not found in file.

As with SEARCH, if the <text> is omitted, the text last used on a SEARCH or NEXT command will be used. The NEXT command is identical to using the Next Match key in Screen Mode, except that it permits the specification of new <text>.

4.1.3.3. REPLACE (R) - Replace text

REPLACE <text>

The REPLACE command allows interactive replacement of one set of text with another. The text given on the REPLACE command is the "old string". Occurrences of it will be searched for exactly like the SEARCH command. The search for occurrences to replace will start at the top of the file unless the SBLK mode is on, in which case the search will start at the top of the marked block of text and be limited to its extent. After entering the REPLACE command, a question mark will appear as a prompt for the "new string" which is to replace occurrences of the old string in the file. If SEARCHFOLD mode is on and you wish the automatic matching of case to proceed as described above, the new string should be entered in lower case. Window will convert lower case letters to upper case according to the SEARCHFOLD rules. It will never change an upper case letter to lower case.

As each occurrence of the old string is found in the text, the line containing it will be printed on the terminal and the prompt:

Line <number> Action (Y, N or Q) ?

will appear. The <number> printed will be the line number within the file where the old string was found. At this time, you should enter one of the three specified letters, which have the following effects.

- Y The occurrence of the old string will be changed to the new string and REPLACE will search for the next occurrence of the old string.
- N The occurrence of the old string will be left unchanged. REPLACE will search for the next occurrence of the old string.
- Q Quit. Leave the occurrence of the old string unchanged and return to Command Mode, terminating the REPLACE command.

4.1.3.4. GLOBAL - Global replacement

GLOBAL <text>

The action of the GLOBAL command is identical to that of the REPLACE command, except that all occurrences of the old string are changed to the new string: the user is not asked to confirm each change. This

Window User Guide

command must be used with great care, as it is very easy to make unintended changes. For example, if you were changing the string "tory" to "conservative", the GLOBAL command would change the word "Conservatory" to "Conservaconservative", which is probably not what you had in mind. As a result, the GLOBAL command is most often used in conjunction with SBLK mode, which limits its effect to only the marked block of text where its results can be better anticipated and verified.

4.1.4. Block commands

The following commands perform various operations on the block of text marked with the Define Block key in Screen Mode.

4.1.4.1. DELETE - Delete marked block

DELETE

All lines in the marked block are deleted from the file. After this command, no block will be marked. Window returns to Screen Mode, with the cursor wherever it was before the DELETE command, except that if it was within the marked block, it will be moved to the first line following the block.

4.1.4.2. COPY - Copy marked block

COPY

All lines in the marked block will be copied to before the line on which the cursor is presently positioned. Note that the cursor must have been moved outside the marked block before the COPY command is given. It is not allowed to copy a block to "within itself", and if this is attempted the message:

Marker error.

will be given. The COPY command leaves you in Command Mode. The marked block is unchanged and still marked, so you may copy it any number of times with successive COPY commands.

4.1.4.3. MOVE - Move marked block

MOVE

The lines in the marked block will be moved from their present position to before the line where the cursor is currently positioned. As for the COPY command, the cursor must be located outside the marked block when the MOVE command is given. As the text is moved from the marked block to its new location, it is deleted, and when the MOVE command is completed, no block will be marked. At the end of the MOVE command, Window will return to Screen Mode with the cursor position unchanged. The action of the MOVE command is identical to that of a COPY followed by a DELETE, but MOVE is much more efficient, especially when moving a large block.

4.1.4.4. CLEAR - Clear marked block

Window User Guide

CLEAR

The marked block will be cleared (unmarked). The lines within the marked block will be unchanged. Following this command, which leaves you in Command Mode, you can go back to Screen Mode and mark a new block if you desire.

4.1.5. Formatting commands

The following commands provide simple formatting of text. They are most useful when using Window to prepare simple documents. WORD, Marinchip's comprehensive text formatter, should be used for tasks more complex than those these commands can accomplish.

4.1.5.1. CENTER - Center text on line

CENTER

The line on which the cursor resides will be centered by deleting all leading blanks and then inserting enough blanks so that the remaining text is centered. The cursor position is adjusted so that it resides on the same character as before (unless it was in the leading blanks, of course).

4.1.5.2. FORMAT - Reformat paragraph

FORMAT

Often when editing text, changes are made which cause individual lines to exceed the width of the screen or to become very short compared to other lines. The FORMAT command causes a paragraph of text to be reformatted so that all lines fit on the screen, and all are as long as will fit. This command is especially useful when preparing simple documents with WINDOW; since WORD performs this operation automatically, it is less important when preparing input for WORD (although it must be used when a line longer than 80 characters is created, since WORD cannot process input lines longer than 80 characters).

To FORMAT a paragraph, move the cursor to the first line of the paragraph. The FORMAT command does not care what character on the line the cursor points to. Then enter Command Mode and enter the FORMAT command. Window will start at the line the cursor is on and scan forward to find the end of the paragraph. The end of the paragraph is identified by any of the followings:

1. A blank line.
2. A period (".") in column one of a line.
3. The end of file.
4. A line whose first nonblank character starts in a column different from the first nonblank in the line below the cursor.

Rule 4 sounds complicated, but what it really means is that Window

Window User Guide

looks at the second line of the paragraph to determine the standard indentation for the paragraph (since the first line may have been indented or have "hanging indentation"). All lines after the second line with the same indentation are considered part of the paragraph. If a line is encountered which is either indented or has hanging indentation, the line before it is considered the end of the paragraph.

FORMAT will preserve indentation on the first line, and indent all subsequent lines of the paragraph the same as the second line was indented. If the paragraph is only a single line and FORMAT causes it to expand to multiple lines, the subsequent lines will begin in column one.

Lines produced by FORMAT will contain as many words as will fit in the length set by the WIDTH command (initially the width of the screen). Words which end with ".", "!", or "?" will be followed by two spaces.

When the FORMAT command is completed, the cursor will be positioned at the first nonblank character of the first line of the paragraph and Window will return to Screen Mode.

4.1.5.3. WIDTH - Set line width for formatting

WIDTH <number>

The line width used for the CENTER and FORMAT commands is initially set to the width of the terminal screen. The WIDTH command can be used to set the line width to any desired value. If the <number> is zero or omitted, Window will set the width back to the width of the terminal screen.

4.1.6. Cursor Positioning commands

The following commands move the cursor in various ways.

4.1.6.1. LINE - Move cursor by lines

LINE [+/-]<number>

The LINE command allows the cursor to be moved either to a specified line in the file, or forward or back a specified number of lines relative to the line the cursor was on before the command. If the <number> is preceded by a Plus or minus sign, the cursor will be moved the specified number of lines forward or back in the file. If not, the cursor will be moved to the specified line number. The cursor will be positioned at the leftmost character on the screen on the specified line. If an attempt is made to move the cursor before the first line in the file (line 1), the cursor will be moved to line 1. If the command tries to move the cursor past the end of the file, the cursor will be moved to the first line after the end of the file. The LINE command returns you to Screen Mode.

4.1.6.2. PAGE (P) - Move cursor by Pages

PAGE [+/-]<number>

Window User Guide

The PAGE command allows the cursor to be moved by pages, where a page is equal to a screenful of lines. If the <number> is preceded by a plus or minus sign, the cursor will be moved by the specified number of pages relative to its current position. If not, Window will go to the absolute page specified. The cursor will be positioned at the leftmost character on the screen at the top of the specified page. Pages are numbered from 1. Window returns to Screen Mode following the PAGE command.

4.1.6.3. PUSH - Save current position

PUSH

Sometimes when editing it is handy to be able to save your current position, go off and look at another part of the file and possibly make a change, then go back to where you were and continue what you were doing before. The PUSH command "pushes" the current position of the cursor on a stack which can hold up to ten such positions. After performing the PUSH command, Window returns to Screen Mode, and you can go off and do whatever you wished to do, knowing that your position has been saved. If you attempt to save more than ten positions with the PUSH command, Window will discard positions starting with the oldest to accommodate the new positions.

4.1.6.4. POP - Return to previous position

POP

The POP command restores the cursor position to that saved most recently with the PUSH command. Window returns to Screen Mode, with the display and cursor position exactly where they were when the PUSH was done (of course, any changes that have been made may have affected the relative cursor position). If no position has been saved with a PUSH command, the POP command will have no effect and the command prompt will be repeated.

4.1.7. Mode switch commands

The following commands turn various modes which affect Window in both Screen and Command Mode on or off. The function of each mode is explained in the following sections. All mode switches are initially off when Window is first called. To turn a mode on, simply type its name followed by "TRUE" when the command prompt is given ("TRUE" may be abbreviated to "T"). To turn a mode off, type its name followed by "FALSE" (which can be abbreviated to "F"). For example, if there were a mode called ZORCH, it would be turned on by:

```
ZORCH TRUE  
ZORCH T
```

and turned off by:

```
ZORCH FALSE  
ZORCH F
```

4.1.7.1. INDENT - Automatic indentation

INDENT TRUE/FALSE

The INDENT mode switch controls alternate action for several command keys in Screen Mode. These alternate actions are suited for editing structured programming languages, which are normally indented to illustrate program structure. If INDENT mode is off, all command keys in Screen Mode function as described in the chapter on Screen Mode. If INDENT mode is on, the action of the following keys differs from the normal action.

Carriage Return

Carriage Return will position the cursor to the first nonblank character of the next line rather than the first character of the next line. If Entry Mode is on, or a new line is being created at the end of the file, Carriage Return will place the cursor on the new line immediately below the first nonblank character on the preceding line. If you desire a different indentation on the new line, you may then simply move the cursor with any of the keys provided for that purpose before entering the new line.

Beginning of Line

The Beginning of Line key will place the cursor over the first nonblank character on the new line, rather than the first character of the line.

4.1.7.2. SBLK - Limit searches to marked block

SBLK TRUE/FALSE

The SBLK switch limits the search and replace commands to act only within the block of lines defined with the Define Block key in Screen Mode. See the description of the search and replace commands above for a complete discussion of the effects of the SBLK mode switch.

4.1.7.3. SEARCHFOLD - Match letters regardless of case

SEARCHFOLD TRUE/FALSE

The SEARCHFOLD mode switch causes the search and replace commands to match letters regardless of whether they are upper or lower case. In addition, it activates a set of rules in the replace commands which attempt to preserve the context of the word being replaced. See the description of the search and replace commands above for a complete discussion of the SEARCHFOLD mode and its impact on each of the commands.

4.1.7.4. TEXT - Allow next/last word between lines

TEXT TRUE/FALSE

If the TEXT mode switch is turned on, the Next Word and Previous Word keys in Screen Mode will be allowed to go to the next line or previous line, respectively, if the end of the current line is reached. Otherwise, those keys will have no effect if the end of line is reached.

Window User Guide

4.1.7.5. WRAP - Move words between lines

WRAP TRUE/FALSE

If WRAP mode is on, and you are typing text at the end of a line, and you run off the right side of the screen, Window will move the word down onto a newly created line to prevent running off the end of the screen. The newly created line will be indented the same amount as the line of which you ran off the end. Note that WRAP mode has this effect only when actually entering characters at the end of a line. If you insert characters in the middle of a line and hence cause words to run off the right side of the screen, WRAP mode will not help you. It is intended primarily for rapid entry of text without having to worry about the width of the screen.

4.1.8. Miscellaneous commands

The following commands don't fit neatly into any of the above categories, so they're grouped together here.

4.1.8.1. HELP - Print command summary

HELP

HELP prints a full-screen summary of Window's Screen Mode and Command Mode commands. This summary is not intended to replace this manual, but just to jog your memory when you've forgotten an infrequently used command. At the end of the HELP display, the message:

Press return when ready:

will appear. When you press Carriage Return, the HELP display will vanish and the Command Mode display will return.

To conserve memory and to allow the HELP display to be modified for various terminals, Window reads the HELP display from a file named WINDOW.HLP. Window looks for this file in the same manner as it looks for the Window Configuration File. Refer to the chapter on the Configuration File for details. If no WINDOW.HLP file can be found, the HELP command will be ignored.

4.1.8.2. BLANKS - Insert blank lines

BLANKS <number>

The BLANKS command inserts the specified number of blank lines before the line containing the cursor. The cursor position is unchanged. This command is useful to create a large "white space" which you intend to fill in with a table or other formatted data. It is generally easier to construct a table when you can put the space for it on the screen and move around in it rather than entering each line in Line Entry Mode.

4.1.8.3. EDIT (E) - Return to Screen Mode

EDIT

Window User Guide

The EDIT command returns you to screen mode. It is identical to simply typing a Carriage Return when the command prompt appears.

Window User Guide

5. The Window Configuration File

When Window is called, it will look for a Configuration File which contains commands to modify its assumed operating modes. The configuration file always has the name:

WINDOW.CFG

and Window will look for it in the following places.

1. In the directory in which the output file for this editing session resides.
2. In the user's assumed directory (NOS), or the system volume, 1/ (Disc Executive).
3. If NOS, in the directory 1:SOURCE.

Window searches the places listed above in the order given, and takes the first WINDOW.CFG file it finds. The order of the search allows you to have three levels of default parameters for Window (two for the Disc Executive). The copy of WINDOW.CFG in 1:SOURCE represents the system wide standards, which are used by all users who do not set up their own parameters. The copy in the user's assumed directory represents that user's preferred operating modes. If special modes are desired when working on a specific program, the user default modes can be overridden by placing a configuration file in the directory where the program resides.

In Disc Executive, there is no need for system wide standards, since there is only one user. Under Disc Executive, the user normally puts his preferred configuration on the system disc (Unit 1), and can override it for a specific program by putting a configuration file on the volume where the program resides (which is normally mounted on Unit 2).

The configuration file consists of statements which set operating modes for Window. Most of these statements are the same as the mode commands in Command Mode, but there are additional statements available in the configuration file which provide special features and allow the configuration to differ based on the type of file being edited.

5.1. Mode statements

The most common use of the configuration file is to set up Window modes. Using the configuration file saves the user from having to enter Command Mode and enter all the mode commands each time Window is called. The mode statements accepted in the configuration file are:

```
INDENT TRUE/FALSE
SBLK TRUE/FALSE
SEARCHFOLD TRUE/FALSE
TAB <number>,<number>,...
TEXT TRUE/FALSE
WIDTH <number>
```

WRAP TRUE/FALSE

These statements have the same format and work exactly like the commands with the same names in Command Mode. Refer to the chapter describing Command Mode for a complete description of these statements.

5.2. GOCMD - Define postprocessor

GOCMD <text>

The GOCMD statement defines the command to be executed to process the output of the editing session when the GO command is used in Command Mode. The <text> on the GOCMD statement will be passed to the operating system as a command, exactly as if it were typed on the user terminal. Two special characters may be used in the <text> to allow the file name being edited to be incorporated in the command:

- % This character is replaced with the entire output file name.
- & This character is replaced with the output file name with the last period and any characters following it deleted.

For example, if we were editing a series of QBASIC programs with names like SORT.BAS, MERGE.BAS, and so on, we might place the following statement in the configuration:

GOCMD QBASIC &.REL=%

If we called Window to edit SORT.BAS and save the GO command, Window would write the output back to SORT.BAS, then execute the system command:

QBASIC SORT.REL=Sort.BAS

which would perform the compilation of the output file.

5.3. PAGE1 - Specify first paging file

PAGE1 <file name>

The PAGE1 statement specifies the name of the file which will be used to hold data from the file being edited if the file is too large to keep in memory. The file name given may be any valid file name for the operating system under which Window is being used. If Window is running under NOS/MT, the file named on the PAGE1 statement will be automatically created if the file being edited exceeds the size of memory. If Window is running under the Disc Executive, the file named on the PAGE1 statement must have been previously created by the user via the CREATE command. If no PAGE1 statement is given in the configuration, the file TEMP1\$ will be assumed. If a PAGE1 statement is given with no file name, the paging feature will be disabled and Window will abort if the size of the file being edited exceeds the size of memory.

5.4. PAGE2 - Specify second paging file

PAGE2 <file name>

The PAGE2 statement specifies the name of the file which will be used to hold data from the file being edited if both memory and the file named on the PAGE1 statement are full. The file specified on the PAGE2 statement is used exactly like the file on the PAGE1 statement, except that it will not be used unless the PAGE1 file has first been filled. If no PAGE2 statement is given, Window will assume TEMP2\$. If a PAGE2 statement is given with no file name, no second paging file will be available, and if the first paging file overflows, Window will abort.

5.5. IF and ENDF - Conditional configuration

IF <type>
ENDF

The IF and ENDF statements are used to make Window's configuration adapt automatically based on the type of the file being edited. For example, we might want one set of tab stops for Pascal, another for QBASIC, and still another for Assembly language programs. For Pascal programs, we may wish INDENT mode, but don't want that mode when editing other types of files.

Variable configurations like this can be achieved by using the IF and ENDF statements to cause configuration statements to be executed only for certain program types. The program type is assumed to be indicated by the characters following the last period in the file name. For example, in the following file names:

UTILSUBS.ASM
GLUPDATE.BAS
EYEHAND.PAS

the program types are ASM, BAS, and PAS respectively.

When an IF statement is encountered, Window will compare the type of the file being edited (if the input and output file are different, the output file type will be tested unless there is no output file, in which case the input file type will be used) with the type named on the IF statement. The comparison considers upper and lower case letters to be the same. If the type matches, the configuration statements following the IF will be executed. If the type fails to match, all statements up to the next ENDF statement will be skipped.

If no <type> follows the IF, the statements following it will be executed only if no type appears in the file name (that is, there is no period in the name). There is no restriction on the length of the <type> used in a file name or on the IF statement.

5.6. Sample configuration file

The following is a typical Window configuration file.

SEARCHFOLD TRUE
IF ASM
TAB 11,21,41

Window User Guide

```
GOCMD ASM &.REL=%  
ENDF  
IF BAS  
  TAB 9,12,15,18,21,24,27,30  
  GOCMD QBASIC &.REL=%  
ENDF  
IF PAS  
  INDENT TRUE  
  TAB 5,10,15,20,25,30,35,40,45  
  GOCMD PASCAL(%,.&)  
ENDF  
IF WRD  
  TEXT TRUE  
  GOCMD WORD PRINT.DEV=%  
ENDF
```

Window User Guide

6. Editing large files

Many text editors limit the size of the file being edited to the size of computer main memory, or require the user to explicitly segment the file being edited into pieces that fit into memory. This is very inconvenient and hard to learn, so Window provides an automatic paging mechanism that transfers portions of the file being edited to temporary files if the file being edited is too large to all fit in memory at the same time. The disc paging mechanism is automatically activated if the file being edited exceeds the size of memory, and requires no user intervention to operate correctly. The paging system attempts to select lines for removal based on the probability they will be referenced, so that the lines most likely to be examined or changed by the user will remain in memory and will be available without delay.

The paging mechanism in Window is set up to work in a manner suitable for most users. Statements in the configuration file may be used to alter the operation of the paging mechanism to adapt it to special configurations or requirements.

6.1. Default paging files

If no PAGE1 or PAGE2 statements appear in the configuration file (see the description of these statements in the chapter describing the Configuration File above), Window will use the files TEMP1\$ and TEMP2\$ as the two paging files. If Window is being used under the Disc Executive, the files TEMP1\$ and TEMP2\$ must have been previously created on the system disc (the disc in Drive 1) before Window is called. If Window is being used under the Network Operating System, files named TEMP1\$ and TEMP2\$ will be created, as needed, in the user's current directory. Any previously existing file with that name will be deleted, and any paging file created by Window will be deleted at the end of the editing session under NOS. Under Disc Executive, the paging files will be left on the disc so they may be used in subsequent editing sessions.

If the file being edited exceeds the size of memory, blocks of the file will be written out to the first paging file, TEMP1\$. If all the blocks written out (which may be less than the size of the file being edited, since memory will still be filled with data from the file) fit within TEMP1\$, TEMP2\$ will not be used. If TEMP1\$ overflows, the second paging file, TEMP2\$, will be opened, and data will be placed into it also.

Window compresses data stored in memory and in the paging files to conserve space. This compression is totally transparent to the user, and has no effect on the editing of the file or the manner in which it is written out at the end of the editing session.

6.2. User-defined paging files

By placing PAGE1 and PAGE2 statements in the Window Configuration File the user may specify files to be used instead of TEMP1\$ and TEMP2\$. If the PAGE1 or PAGE2 statement names no file, that level of paging will be turned off and Window will abort if the file being edited

Window User Guide

requires the level of paging thus disabled.

Under the Disc Executive, a common use of the PAGE1 and PAGE2 statements is to place the paging files on Drive 2, along with the file being edited. This is done when using a system disc which contains so many programs that there is not room for TEMP1\$ and TEMP2\$ files large enough for the file being edited.

Under the Network Operating System, there is no reason to use a PAGE2 file unless the PAGE1 file is contiguous or the PAGE2 file is placed on a different volume from the PAGE1 file. Since files may expand as they are written, the PAGE1 file can overflow only if all space on the volume where it resides is exhausted. In this case, a user-specified PAGE2 file on a different volume can contain the overflow. Another possible use for PAGE1 and PAGE2 files under NOS might be to create a contiguous PAGE1 file on a very fast paging device (for example, a head-per-track disc), then use a regular PAGE2 file on a larger but slower device which could expand to hold any overflow that could not fit on the fast device.

6.3. Paging error conditions

If the size of the file exceeds the size of main memory and Window is unable to remove lines to the paging files because they have not been configured, cannot be opened, are filled to capacity, or cannot be written into, or Window cannot bring lines back from a paging file because of a read error, Window will abort and print a message on the user terminal describing the nature of the error and giving the name of the paging file in which the error occurred.

Window User Guide

7. Generating and Configuring Window

This chapter explains how to generate Window and how to configure it for a specific terminal. If you already have a copy of Window properly configured for your terminal, this information need not concern you; it is intended only for those who wish to generate versions of Window for their own use or for use by others, and for users who wish to adapt Window to a terminal not supported by Marinchip Systems.

The instructions given below are in "cookbook" form. That is, they do not dwell at length on exactly what is done by each command. If you wish simply to generate Window, the instructions given should be sufficient. If you wish to develop your own terminal drivers, examination of the drivers we supply will yield far more information than any discussion here could provide.

7.1. The Window release discs

Window is supplied on four discs, labeled:

- Window Execution
- Window QBASIC Master
- Window Drivers
- Window Relocatables

These are Disc Executive format, single density diskettes. The QBASIC Master disc can be supplied only if you have purchased QBASIC and provide us with your QBASIC serial number when you order Window. It is necessary to generate Window. Note that you must have QBASIC only to generate Window; hence, if you order a version of Window for a terminal that we support, you need not purchase QBASIC.

The entire Window generation process is done under the Disc Executive, even if the version of Window being generated is to be used under the Network Operating System. Window automatically adapts to Disc Executive or NOS, so a given version may be used on either system without change.

7.2. Selecting an input driver

The first step in generating Window is to choose a terminal input driver and compile it. We supply a standard terminal input driver, TERMIN-STD, as well as special terminal input drivers for various popular CRT terminals. The terminal input driver translates the codes generated by the terminal's keys into the commands Window executes when a key is pressed. If TERMIN-STD is used, Window will respond to the keys listed in the sections on editing commands in the Screen Mode chapter, and in the description of the standard terminal input at the end of this manual. Other terminal input drivers are used to make Window respond to the codes generated by cursor positioning keys and special function keys on those terminals. Refer to the terminal input tables at the end of this manual for information on how various terminals are handled. Remember, you can always use TERMIN-STD. You should use a special driver for a terminal if you want to use its special keys.

Window User Guide

The terminal-specific input drivers all have names which begin with TERMIN- and end with a mnemonic for the terminal. For example, the input driver for Hazeltine terminals is TERMIN-HAZ. All terminal input drivers are on the Window Drivers disc. Look at the directory of that disc and note all files with names beginning with "TERMIN-". If you wish to develop your own terminal input driver, examine the ones we supply and use them as a model for your own development.

Once you have chosen a terminal input driver, you must copy it to the Window Relocatables disc and then compile it. To do this, perform the following steps. After any step where you are asked to type something, wait for the operation to complete and the appropriate prompt to appear before going on to the next step. If you get an error message at any step, correct the problem before proceeding.

1. Place Window QBASIC Master in Drive 1
2. Place Window Drivers in Drive 2
3. Type EDIT
4. Type `RF 2/TERMIN-xxx` where xxx is desired driver
5. Remove Window Drivers disc from Drive 2
6. Place Window Relocatables in Drive 2
7. Type `WF 2/TERMIN.BAS`
8. Type AB
9. Type `QBASIC CONS.DEV=2/TERMIN.BAS,TEMP1$`,
10. Remove Window Relocatables from Drive 2
11. Place Window Drivers in Drive 2
12. Type `2/QBOPT`
13. Remove Window Drivers from Drive 2
14. Place Window Relocatables in Drive 2
15. Type `ASM 2/TERMIN.REL=TEMP1$`

After completing these steps, the correct driver will have been placed on the Window Relocatables disc for later use in the generation process.

7.3. Selecting an output driver

Next it is necessary to choose and compile the output driver for the terminal Window will be used with. Because each terminal has a different set of codes for the various functions Window uses, it is necessary to have a driver written specifically for each terminal; there is no standard terminal output driver because there is no such thing as a standard terminal.

All of the terminal output drivers are on the Window Drivers disc, and have names that begin with TERMOUT-. For example, TERMOUT-HAZ is the terminal output driver for the Hazeltine terminals, and TERMOUT-IBM is the driver for the IBM 3101. Find a driver for your terminal, or, if none is supplied, write a driver using the ones we supply as a starting point. If you use one of our standard drivers, be sure to look at it with the editor and see if it contains a model number variable. For example, the Hazeltine driver has a variable called HAZMODEL, which should be set to the model number of the terminal being used (for example, 1500, 1420, etc.). Some drivers must know the terminal model number because features present in terminals differ from model to model.

Window User Guide

After having chosen the terminal output driver, it must be copied to the Window Relocatables disc and compiled. The procedure for doing this is similar to the procedure we used with the terminal input driver, and consists of the following steps.

1. Place Window QBASIC Master in Drive 1
2. Place Window Drivers in Drive 2
3. Type EDIT
4. Type RF 2/TERMOUT-xxx where xxx is desired driver
5. Remove Window Drivers disc from Drive 2
6. Place Window Relocatables in Drive 2
7. Type WF 2/TERMOUT.BAS
8. Type AB
9. Type QBASIC CONS.DEV=2/TERMOUT.BAS,TEMP1\$,
10. Remove Window Relocatables from Drive 2
11. Place Window Drivers in Drive 2
12. Type 2/QBOPT
13. Remove Window Drivers from Drive 2
14. Place Window Relocatables in Drive 2
15. Type ASM 2/TERMOUT.REL=TEMP1\$

As with the terminal input driver, wait for each step to finish before moving on to the next step; if you get an error message from any step, resolve the problem before continuing.

7.4. Linking Window

After the drivers have been compiled, all that remains is to link the various modules comprising Window into the executable program. This is accomplished by placing the Window QBASIC Master disc in Drive 1 and the Window Relocatables disc in Drive 2 and typing the command:

```
LINK TEMP1$=@2/L
```

The Linker will read in all the Window files and place a copy of Window in the file TEMP1\$. When the Link completes, remove the Window Relocatables disc from Drive 2 and install the Window Execution disc. Copy Window to the Execution disc with the command:

```
BCOPY 2/WINDOW=TEMP1$
```

When this command is done, a ready to use copy of Window will reside on the Window Execution disc.

7.5. Installing Window in your system

For Disc Executive, installing Window simply involves copying it to your system disc. To find out how big a file is required, you can type the command:

```
SIZE 2/WINDOW
```

after the BCOPY, and the size of the file, in sectors, will be printed. Since Window is quite large, you may choose to make up a system disc for editing, which has only SHELL\$.OBJ, WINDOW, and the TEMP1\$ and TEMP2\$ files. Since Disc Executive permits you to change discs at any time, you can install this disc when editing, the put in

Window User Guide

the regular system disc, or the Pascal or QBASIC disc, when ready to compile or otherwise process the file just edited.

For systems using the Network Operating System, the copy of Window on the Window Execution disc should be converted to NOS format with the CONVERT command as described in the NOS manual. Once converted, Window should be placed in 1:BIN so that all users may call it by simply typing:

```
WINDOW <file name>
```

If a system running NOS has several different types of terminals, then each user must have his own copy of Window properly configured for his terminal. The correct copy of Window can be placed in the user's assumed directory, so that the simple command, "WINDOW" will call the correct version for his terminal.

Finally, the file used by the HELP command, WINDOW.HLP, should be installed. This file is supplied on the Window Execution disc, in source form, and should simply be copied to the system disc (for Disc Executive) or into the directory 1:SOURCE (for NOS). If you wish to change the text printed by the HELP command (for example, because you use different control characters for the Screen Mode commands), you can edit the WINDOW.HLP file and make whatever changes you desire.

7.6. For more information on drivers

Because terminal input and output drivers for Window are constantly being developed for new terminals, and existing drivers are being revised to accommodate new models of terminals, it is not possible to provide complete descriptions of the various drivers in this manual. The file DRIVERS.DOC on the Window Drivers disc is an informal set of notes describing the actual drivers provided with a release of Window. List that file for information on the drivers we supply.

Window User Guide

8. Terminal reference guides

The following sections describe the keys used for the various Screen Mode commands for specific terminals. Unless a special driver has been provided for your terminal, you should use the standard driver reference given below.

The drivers for specific terminals have been designed wherever possible to permit the standard keys to be used in addition to the special keys provided by the terminal. Where incompatibilities between terminals require a change in one of the standard command keys (hence the standard key will not work on that terminal and another key MUST be used), the command will be flagged with "**" in the table for that terminal.

Window User Guide

8.1. Standard terminal input

Cursor Right.....	Control L
Cursor Left.....	Control H or Backspace
Cursor UP.....	Control K
Cursor Down.....	Control J or Linefeed (LF)
Carriage Return.....	Control M or Return (CR)
Tab.....	Control I (TAB or HT)
Beginning of Line.....	Control U
End of Line.....	Control N
Home.....	Control ^ or Control Shift N
File End.....	Control E
Page UP.....	Control R
Page Down.....	Control T
Center Cursor.....	Control S
Next Match.....	Control X
Insert/Split Line.....	Control B
Delete Line.....	Control Z
Concatenate Lines.....	Control O
Erase Line.....	Control DEL or Control Shift O
Delete to End of Line.....	Control Y
Next Word.....	Control A
Previous Word.....	Control W
Delete Word.....	Control V
Insert Character.....	Control F
Delete Character.....	Control D
Erase Character.....	Rubout or DEL
Escape.....	ESC or ALT MODE or Control [
Character Insert Mode.....	Control Q
Entry Mode.....	Control \ or Control Shift L
Define Block.....	Control P

Window User Guide

8.2. Hazeltine terminal input

Cursor Right.....	Right Arrow or Control P
Cursor Left.....	Left Arrow or Backspace
Cursor Up.....	UP Arrow or Control K
Cursor Down.....	Down Arrow or LF
Carriage Return.....	CR
Tab.....	TAB or Control I
Beginning of Line.....	Control U
End of Line.....	Control N
Home.....	Home or Control Shift N
File End.....	Control E
Page Up.....	Control R
Page Down.....	Control T
Center Cursor.....	Control S
Next Match.....	Control X
Insert/Split Line.....	Control B
Delete Line.....	Control Z
Concatenate Lines.....	Control O
Erase Line.....	Control Shift O
Delete to End of Line.....	Control Y
Next Word.....	Control A
Previous Word.....	Control W
Delete Word.....	Control V
Insert Character.....	Control F
Delete Character.....	Control D
Erase Character.....	Rubout
Escape.....	ESC
Character Insert Mode.....	Control Q
Entry Mode.....	Control Shift L
** Define Block.....	Control L

Window User Guide

8.3. IBM 3101 terminal input

Cursor Right.....Right Arrow or Alt L
Cursor Left.....Left Arrow or Backspace
Cursor Up.....Up Arrow or Alt K
Cursor Down.....Down Arrow or Alt J

Carriage Return.....Return
Tab.....Tab or Alt I
Beginning of Line.....Alt U
End of Line.....Alt N

Home.....Home or Alt -
File End.....PF3 or Alt E
Page Up.....PF1 or Alt R
Page Down.....PF2 or Alt T
Center Cursor.....Clear or Alt S
Next Match.....Alt X

Insert/Split Line.....PF6 or Alt B
Delete Line.....Alt Z
Concatenate Lines.....Alt O
Erase Line.....Erase Input or Alt /
Delete to End of Line.....Erase EOL/EOF or Alt Y

Next Word.....Alt A
Previous Word.....Alt W
Delete Word.....Alt V

Insert Character.....PF4 or Alt F
Delete Character.....PF5 or Alt D
Erase Character.....Del

** Escape.....Erase EOS
Character Insert Mode.....PF8 or Alt Q
Entry Mode.....PF7 or Alt =
Define Block.....Alt P