A New Word Processor for the Color Computer 3 OS/9

Window Writer

designed by
Raj Dash
published by

OWL-WARE

Please Note:

Registration Number:_____

This registration number has been assigned to:_____

This disk is protected by the "Scare-Ware" policy of OWL-WARE. We do not copy protect our products. We do make full use of all legal protection of our copyright. This protection includes, but is not limited to, offering rewards for illegal copies of our software, publishing the names of illegal users, and taking court action against illegal users. If you have an illegal copy of our programs, call us to see if you are eligible for a reward. If you are the first to turn in a copy made from an illegal disk, you will receive either a portion of court settlement or a gift certificate at our option.

As the owner of *Window Writer*, you may make as many copies of the original disk as you feel are necessary to protect your property. Each copy made from the original will retain your specific code sequence which makes it traceable. You may use these copies on any Color Computer 3 which you normally use and on both hard drive and floppy systems. You MAY NOT GIVE OR LEND THE PROGRAM TO ANY OTHER USER. If you do give a copy to another user, you are responsible for all copies which may ultimately result from your original as well as legal fees necessary to enforce this agreement. If a copy is lost or stolen, notify us at 215-837-1917. This notification is necessary to prevent legal action being taken against you if illegal copies of your version enter the market place.

Table of Contents

Chapter 1.

INTRODUCTION

OVERVIEW

Window Writer is unlike any word processor which has ever been available for the Color Computer. It is one of the first applications for the Color Computer 3 to take full advantage of the powerful OS/9 operating system and the advanced hardware of the Color Computer 3. The unique features of OS/9 make it possible to create a word processor which is as modern and professional in action as any available for the IBM systems and the Mac while still having a fast, snappy response. With *Window Writer* it is possible to edit several files at the same time, to print one file while editing another, or even to run other programs while using *Window Writer*.

Like most modern word processors, there is always more than one way to access any editing feature or help screen. You can access pull down menus from mouse (or joystick), "keyboard mouse," or "quick keys." You also can use control keys to directly access many of these features. Full help screens are quickly available for all editing features. These can remain visible on the screen or be quickly removed to get back to full screen editing capability.

The innovative features of *Window Writer* include:

- An easy-to-use yet powerful 80 column editor with the ability to copy, delete, and move words, lines, and paragraphs at one time.
- There is a formatting module versatile enough to handle the most complex tasks, yet so easy to use that it makes the simple jobs even simpler.
- All formatting codes and special printer codes are either invisible or represented by colored or underlined text. Thus the width of the text is exactly as displayed on the screen.
- An easily customized printer driver allows Window Writer to use over 60 intelligent printer features, all of which are accessible from within the editor via menu selection.

- A "print preview window" is provided so that the you will be able to see what the printed document will look like before it is printed - including headers, footers, page numbers, etc.
- Pull-down menus are fully implemented so that you need not memorize various control keys. However, if you prefer the standard keyboard control interface, it is still there.
- Help menus are only a keystroke away (so you will need to make only minimal reference to the instruction manual).
- The ability to edit multiple files, edit one file while printing another, or perform various other tasks simultaneously is an extremely valuable asset provided by the OS/9 operating system.
- This software comes fully configured, but can be completely user configured for every feature including menu colors and contents.

This manual provides all the necessary information needed to utilize all the features of *Window Writer*. It will most likely be necessary to read only the first few pages of this manual before getting started, however. Please read this chapter and the first 3 parts of the next Editor chapter (Chapter 2) before beginning. Feel free to experiment as that is probably the easiest and fastest way to learn!

SYSTEM REQUIREMENTS

Window Writer is designed to be a professional word processor in both capabilities and function. Due to its intended functions, the minimum system requirements for *Window Writer* are higher than for other Color Computer word processors.

MINIMUM REQUIREMENTS

The minimum system requirements for a *Window Writer* system are:

- Color Computer 3
- OS/9 Level II
- 1 Single-Sided Floppy drive system
- 80 Column Monitor (Color or Monochrome)

RECOMMENDED SYSTEM

To get full usage of all of the *Window Writer* features, the following additional equipment is recommended in order of priority:

- 2 Floppy Drives (or at least 1 double sided drive)
- 512K Upgrade
- Color Monitor (80 Column - either RGB or composite)
- Printer
- Mouse or Joystick

OPTIONAL SYSTEMS ADDITIONS

The following equipment will make it easier and nicer to use, but are certainly not required:

- High-Res Joystick Input
- Additional Floppy Drives
- Multi-View
- Hard Drive

MAKING A BACKUP

Before doing anything else, put a write protect tab on your original *Window Writer* disk. The first order of business is to make a working copy of this *Window Writer* program disk. If your computer is off, turn it on and boot up your standard OS/9 system disk as outlined in the OS/9 manual. You must first format a blank diskette that will hold the backup of the data on the *Window Writer* diskette. To do this insert your OS/9 system diskette into drive /d0 and type:

```
FORMAT /d0 '35'
```

Remove the system disk from /d0 and insert an unused diskette into drive /d0. Continue with the below instructions.

SINGLE DRIVE OPERATION

If only one disk drive is available, you must perform a single drive backup. You will be prompted to "Ready Source" and "Ready Destination" disks. The source disk is the *Window Writer* diskette and the destination disk is any formatted diskette. You will have to swap these several times. With the OS/9 System disk in drive d0, type:

```
LOAD /d0/CMDS/Merge
LOAD /d0/CMDS/Attr
LOAD /d0/CMDS/Backup
```

Put the *Window Writer* program disk in the disk drive (drive d0) and at the system prompt (OS9:) type the following series of commands:

```
BACKUP /d0 s #24k
```

When the OS9 prompt returns, type:

```
UNLINK Backup
```

The original *Window Writer* disk has now been backed up. Put the original disk away in a safe place and use the backup from now on. Put the backup disk just created into drive /d0 and type:

```
CHD /D0
CHX /D0/CMDS
BuildWW.1
```

Follow the prompts until the copying is complete. You will be instructed to exchange the *Window Writer* program disk (the backup copy) with a copy of your OS/9 Boot/Basic09 disk - so have it ready. *Window Writer* will merge the 'RunB' file on the Basic09 disk with some files of its own. The new file that is created (on the *Window Writer* Disk) will still be called 'RunB' and can be used to run other programs that you may have. However, in order to run *Window Writer*, this *modified* RunB must be the one that is used.

DOUBLE DRIVE OPERATION

If two disk drives are available, you may perform a double drive backup. With your OS/9 System disk in the primary disk drive (drive /d0), type:

```
LOAD /d0/CMDS/Merge
LOAD /d0/CMDS/Attr
LOAD /d0/CMDS/Backup
```

Put the *Window Writer* program disk in the first disk drive (drive /d0) and put a blank formatted disk in the second disk drive (drive /d1). At the system prompt, (OS9:), type the following series of commands:

```
BACKUP /d0 /d1 #24k
```

When the OS9 prompt returns, type:

```
UNLINK Backup
```

The original *Window Writer* disk has now been backed up. Put the original disk away in a safe place and use the backup from now on. Put your Boot/Config/Basic09 disk into drive /dd and type:

```
CHD /d1
CHX /d1/CMDS
BuildWW.2
```

Follow the prompts until the copying is complete. *Window Writer* will merge the 'RunB' file on the Basic09 disk with some files of its own. The new file that is created (on the *Window Writer* Disk) will also be called 'RunB'and can still be used to run any other programs that you may have. However, in order to run *Window Writer*, this *modified* RunB must be the one that is used.

GETTING STARTED

512K SYSTEMS

If you have 512K, you will probably want to use a RAM disk (described later) to store text. This will greatly improve the performance of *Window Writer*. There are two files that should be copied into the MODULES directory of the Boot/Config/Basic09 disk. See your OS/9 manual for details on this process. They are 'ram.dr' and 'r0.dd'. These files are in the CMDS directory of the *Window Writer* diskette. Once they are copied, use the config utility to make a new bootfile. Be sure

to select 'ram.dr' and 'r0.dd'. Also be sure to select one of the window interface modules, GRFINT or WINDINT (**not VDGINT**).

If you do not want to create a new boot disk, you may simply load the RAM disk into memory (this wastes a great deal of system memory). With the *Window Writer* disk in drive /dd type:

```
LOAD /dd/CMDS/RamDisk
```

Regardless of whether you put the Ram disk in a boot file or simply loaded it off disk, you must type (with the *Window Writer* disk in drive /dd):

```
chd /dd
chx /dd/cmds
iniz R0
RDForm 20
```

The 'iniz R0' is absolutely necessary. It initializes the device drive, ram.dr. The 'RDForm' is the RAM disk formatter. The number that follows RDForm tells it the number of 8K blocks of memory to use for the RAM disk. A total of 20 blocks will be used by the Ram disk (160K of memory). You must use a number smaller than 20 if a large number of modules and/or windows are normally used on your system.

If the system is already booted up, put the *Window Writer* disk in drive /dd and type at the OS/9 prompt:

```
chd /dd
chx /dd/cmds
ww
```

If the computer is off, turn it on, (initialize the RAM disk as described above if you want to) insert the *Window Writer* program disk into drive /d0 and type:

```
DOS
```

Once the OS/9 prompt appears follow the above instructions.

Window Writer will first ask you for the device (such as a floppy drive or hard drive) to be used to store the text that you will be entering or editing. Currently only devices /d0 (a floppy drive) and /r0 (a RAM drive) are defined. If more devices are present on your system (such as a second disk drive /d1 or a hard drive /h0, *Window Writer* must be informed that such devices exist (see "Customizing the System" for more information on this). If nothing is entered, *Window Writer* defaults to using the last device listed (/r0 - the RAM disk). This RAM disk acts and is used just as though it were exactly the same as a standard floppy disk drive, except for a few differences. The main advantage of a RAM disk is that it is much faster than a standard floppy drive. The main disadvantage is that the contents of the RAM disk are erased when the computer is turned off. *Window Writer* uses the RAM disk to store the

information that you type. Once you are done typing, *Window Writer* copies what is stored in the RAM disk to your floppy disk so you have a permanent record of the information typed. It is a good practice to make frequent saves during your normal work period.

Window Writer will then ask you for the number of pages worth of information you wish to reserve within the RAM disk itself for *Window Writer*. If you decide to start another copy of *Window Writer* in another "window", it too must share this space on the RAM disk as the first Window Writer program. Thus, if you wish to edit two files simultaneously, you will want to reduce the space that each uses.

You should see a black screen with a grey border in a matter of a few seconds. You are now "in the editor." This is where you will type in all the information you need printed and where changes will be made to the text as will be described in the upcoming sections.

OS/9, being a multi-tasking operating system, has the ability to run more than one program simultaneously. Furthermore, it has the ability to run them in different "windows" or screens. To run two or more *Window Writer* programs simultaneously is not a difficult task as long as there is enough memory available.

The first step is to create a window (with a shell) in which another *Window Writer* program can be run. The OS/9 manual covers this topic in great detail. Once this has been done, proceed as usual to start *Window Writer*.

128K SYSTEMS

If your machine has only 128K of memory, performance will be reduced. No RAM drive can be installed, and thus a floppy drive must be used to store text. This will result in slower operation and reduced storage capacity. You will also not be able to access the shell. Thus you must start up in a slightly more cumbersome fashion. Type:

```
chd /dd
chx /dd/cmds
ex runb ww
```

This will start up *Window Writer* without a shell. The shell module should unlink itself from memory and free up 8K of memory for Window Writer to use. Make sure that the shell is not locked into memory in the startup file of your boot disk. If 'link shell' is present in your startup file, delete it.

Chapter 2.

THE EDITOR

THE CURSORS

1. THE CURSOR & CONTROL KEYS

There are two cursors present on the screen. One is a white colored text cursor. This cursor indicates where text will be inserted. The other cursor is a green mouse cursor. This cursor indicates the position of the mouse at all times.

CONTROL KEYS - <CTRL> & <ALT>

There are essentially two control keys that *Window Writer* recognizes. A control key gives you a way to tell Window Writer that you wish to enter a command (such as reformatting a document or saving a document to disk), rather than insert text which is what *Window Writer* normally expects. The most commonly used control key is unsurprisingly called the Control key, represented by <CTRL> on the keyboard. The other control key is represented by <ALT> on the keyboard. To use a control key sequence, hold it down and type the next key.

When these control keys are used in conjunction with one of the other keys on the keyboard, rather than inserting text Window Writer will perform some task such as reformatting the document, deleting a word, etc. Note that some of pull-down menus can be quickly accessed by holding down the <ALT> key and then typing the control letter of that menu. See Figure 1. for a view of the *Window Writer* screen. There is an alternative way to execute a command using a mouse that you may find easier to learn and use.

THE MOUSE

One of the more exciting features of *Window Writer* is its use of pull-down menus. They can be used so that you need not remember which key combination executes what command. All you need is a

joystick or mouse with a hi-res pack plugged into the right joystick port. To use a low-res mouse or joystick or to use the left joystick port, see "Customizing the System." The keyboard can also be made to simulate a mouse. Hit <CTRL><CLEAR> to enter the "keyboard mouse" mode. Use the arrow keys to move the cursor and use <F1> and <F2> to simulate the two push buttons that are present on a mouse. Note that the arrow keys will no longer function as they usually do nor will the keys <F1> and <F2> function as they normally do. To restore their normal operation, hit <CTRL><CLEAR> once more. This feature is called the "keyboard mouse". Note that if you use this "keyboard mouse" feature, you should hit the <ALT><CLEAR> after leaving *Window Writer* to reset the keyboard.

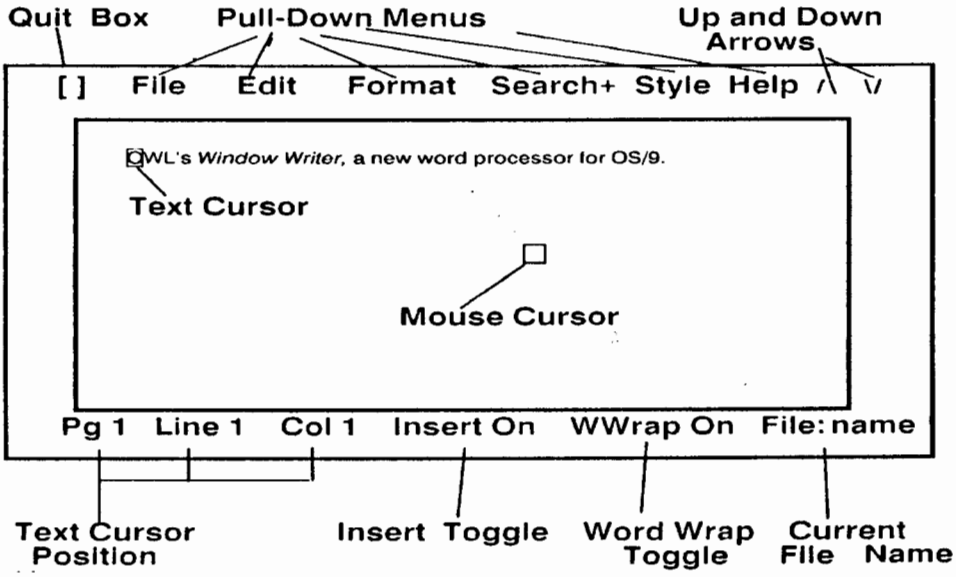

Figure 1. Screen Make Up of *Window Writer*

To activate the menuing system, simply move the green mouse cursor to the menu bar at the top of the screen. Commands are grouped according to their general function. Simply select the grouping desired and hit the button. The menu will appear once the button is released. Again use the joystick or mouse to position the cursor over the desired command and hit the button once more in order to execute the command. It will never be necessary to remember what the control keys do if you use this interface.

THE ARROW KEYS

The arrow keys are the fundamental cursor control keys. They allow you to move a space in any one of four directions at a time, up, down, left, or right. Note that if a key is held depressed for a brief moment, it will automatically repeat. This is extremely convenient to position the cursor quickly. In addition, you may use the <SHIFT> and <CTRL> keys in combination with the arrow keys to move at a greater rate than one character at a time.

MOVE WORD (PAGE) AT A TIME - <SHIFT><ARROW KEY>

By pressing <SHIFT> followed by a left or right arrow key, you may move one word to the left or right respectively. Use <SHIFT> followed by one of the up or down arrow keys to move up or down one page (screen full) at a time.

MOVE LINE(S) AT A TIME - <CTRL><ARROW KEY>

Using <CTRL> with the arrow keys yields slightly more drastic results. If the <CTRL> key is held while hitting either the left or right arrow keys, the cursor will jump either to the beginning or end of the line. If one of the up or down arrow keys is used, the cursor will jump either to the top or end of the document.

MOVE TO SPECIFIC LINE - <CTRL><W>

This is accomplished using the <CTRL><W> command. After typing this command, enter the line number you wish to move to and hit <ENTER>. *Window Writer* will move the cursor to the line specified if possible. If you enter a number that is greater than the number of lines in the text buffer, *Window Writer* will place the cursor on the last line. If you enter a number less than one, *Window Writer* will move the cursor to the first line.

MOVE TO SPECIFIC PAGE - <ALT><->

Window Writer will ask for the page number to move to. Enter a number and hit <ENTER>. *Window Writer* will move the cursor to the top of the specified page if possible. If you enter a page number that is greater than what is in the text buffer, *Window Writer* will move the cursor to the end of the document. If you enter a page number that is less than what is in the text buffer, *Window Writer* will move to the top of the document.

THE MOUSE

It is also possible to move the cursor using the mouse. To do so, simply press the button on the mouse. While this button is depressed, the cursor will be under control of the mouse. You can also use the mouse to move a page (screen full) at a time. Move the mouse up to the menu bar to do this. If you click on the up arrow, *Window Writer* will go back one page. If you click on the down arrow, *Window Writer* will go forward one page. The up arrow and down arrow are represented by (/\) and (\/), respectively on the screen. See Figure 1.

3. INSERTING TEXT

When *Window Writer* is first started, you are placed in the editor and are ready to type. Simply start typing. Type a few lines to get the feel of it. Notice that as you reach the end of the line, the word you are typing may move down to the start of the next line. This is called "word wrap". You should not press < ENTER > at the end of each line as you would on a typewriter.

WORD WRAP - <ALT><W>

This word wrap feature can be turned off if you wish. Whether word wrap is on or off is displayed in the status window at the bottom of the screen. In order to turn word wrap off, hit <ALT><W>. You will see that the status window confirms that word wrap has been turned off. To turn it back on, simply hit <ALT><W> again. Clicking the mouse on the "word wrap" item in the status window also toggles word wrap. See Figure 1.

INSERT/REPLACE MODE - <ALT><I>

When you type, text is normally inserted into the current cursor position. Thus, if you wish to make a spelling correction, for example, you would first have to type in the correct spelling and then delete the incorrect spelling. This can become tedious if there are many corrections to be made. *Window Writer* provides for an alternative. If you hit <ALT><I>, the insert mode will be turned off and the "replace" mode will become activated. Anything you now type overwrites what was there previously. Thus, the old text is replaced with the new. The step of deleting the old text is eliminated. In order to get back into the insert mode, hit <ALT><I> one more time. Clicking the mouse on the "insert" item in the status window also toggles the insert mode. See Figure 1.

4. DELETING TEXT

There are several commands that are available for deleting text. These commands also are accessible from the pull-down menus.

DELETE CHARACTER UNDER - <CTRL><D>

The most commonly used one is the character delete command. In order to delete the character over which the cursor is positioned hit <CTRL><D>. One letter will be deleted where the cursor once was. If the cursor was at the end of the line, part of the line below the current line will be appended to the current line. If the word wrap feature is on, *Window Writer* will try not to break words when appending lines together.

DELETE CHARACTER LEFT - <CTRL><V>

There is also a variation of the <CTRL><D> command. This variation allows you to delete before the cursor instead of after it. Hit <CTRL><V> in order to execute this command.

DELETE WORD RIGHT - <CTRL><F>

It is also possible to delete entire words one at a time rather than characters at a time. By hitting <CTRL><F>, you can delete the word that the cursor is currently positioned at. If the cursor is positioned in the middle of a word, only the right hand portion of the word will be deleted.

DELETE WORD LEFT - <CTRL><B>

The <CTRL><B> command allows you to delete the word before the cursor instead of the one after it.

DELETE LINE RIGHT - <CTRL><G>

You can delete whole lines at one time as well. If you type <CTRL><G>, all the text after the cursor on the current line will be deleted and the following line will be moved up onto the current line if room is available.

DELETE LINE LEFT - <CTRL><N>

The line delete left command will delete all the text up to the cursor on the current line - this is just the opposite of the previously mentioned command <CTRL><G>.

5. FORMATTING COMMANDS

You will soon find that deleting and inserting often yields fragmented text that is not very aesthetically arranged. *Window Writer* has two commands that easily remedy this situation:

REFORMAT PARAGRAPH - <CTRL><A>

This command will realign the paragraph in which the cursor positioned. Feel free to use this command often as it will ensure that the text is properly aligned at all times. Remember that this command only affects the current paragraph – nothing else.

REFORMAT ENTIRE TEXT - <ALT><A>

There is also a command to realign the entire text if necessary. If you decide to change the width of the text in the document, for example, it would be necessary to use this command since the entire text needs to be realigned.

ALTER LINE WIDTH - <ALT><L>

Just mentioned was the fact that the width of the text could be altered. <ALT><L> is the command used to accomplish this. Simply enter a number between 5 and 79 (larger widths can be used when actually printing out to the printer) for the width of the text. Now all alignment commands here on in will use the new width.

ALTER JUSTIFICATION MODE - <ALT><;>

You will be prompted to select one of the following justification modes:

- Left justification means that the text will be made flush with the left hand side of the page.
- Right justification adds spaces before a line so that the right hand margin lines up.
- Center justification centers a line based upon the current line width.
- Full justification inserts spaces between words so that both the left and right sidess are flush with the margins.

Justification may be switched off. Note that extra spaces between words will be deleted. This "undoes" full justification. The align commands (<CTRL><A> or <ALT><A>) should be used to realign those parts of the text to be rejustified or reset to a different line width.

AUTO INDENT - <ALT></>

You will be prompted to decide whether or not auto-indentation is desired. When auto-indentation is in effect, *Window Writer* will position the cursor the selected distance to the right of the left margin whenever you press <ENTER> or whenever word wrap occurs. *Window Writer* will attempt to make the left margin of the new line up with the left margin of the previous line. This mechanism makes it easier for you to create blocks of text that are set off from the rest of the text such as

when you wish to enter an excerpt from a piece of literature. You can also use it as an aid in creating good-looking "structured" programs.

REFORMAT SINGLE LINE - <ALT><G>

Use this command to align only a single line rather than a paragraph or the entire document. This is how single lines may be centered. Simply select the "center" justification mode described above and hit this command to center the current line. Do not forget to turn the justification mode back to whatever it was previously set to.

FIND PAGE BREAKS - <ALT><P>

This command is the page break finder. If the text on the screen is the same width that it will be when printed out, *Window Writer* will accurately determine where page breaks will occur in the text. You will be asked whether the page breaks should be revalidated. Page breaks need to be revalidated in the following cases:

- A new file is opened for editing.
- A New Page command is entered.
- A Top Margin command is entered.
- A Bottom Margin command is entered.
- A Lines per Page command is entered.
- Text lines are inserted or deleted in text that contains one of the above commands.
- Upon returning from the formatting module.

You may deactivate visible page breaks by hitting <ALT><P> once more.

INSERT FORMAT CODE - <CTRL><F2>

This command will cause a "formatting code" menu to appear on the screen. Use this menu to change the way text is formatted. The mouse should be used to select the desired code to be inserted. Then click the mouse button to select that code. More than one code can be inserted at a time. When all the codes have been selected, click the mouse on the "close" box in the upper left hand corner of the screen or hit <BREAK> to get back to the editor. Once the format code has been selected, you may be asked for additional information. For example, if you selected the 'Top Margin' code, you will be asked for the number of lines to be used for the top margin.

However, it may not always be so obvious what *Window Writer* is asking for. For example, if you select the 'New Header' command, you will be prompted to enter the 'Line Number in File'. The number you enter specifies what line in a file called 'header_filelist' (present in your /dd/SYS directory) contains the name of the file which contains the text

to be printed out in the header. For more information on this see the "Formatter" section.

GO TO FORMATTING MODULE - <ALT><E>

In order to print out the text and access all other formatting commands such as lines per page, left margin etc., it will be necessary to type this command to start up the formatting module which is used to handle complex formatting situations. The text currently being edited will not be lost, so feel free to move between the editor and formatter whenever desired. See Chapter 4. on the Formatter.

6. MASS TEXT HANDLING COMMANDS

THE CLIPBOARD

Many of the editing commands mentioned earlier are sometimes too cumbersome to use when dealing with large amounts of text such as a paragraph or several paragraphs. In order to edit larger amounts of text, you should use the clipboard. You can move, copy, delete, or save to disk a marked portion of text with this clipboard. The clipboard is used as a temporary storage area for text. In reality, it is nothing more than a standard disk file. One may mark a portion of text to be sent to this clipboard. The contents of the clipboard may then later be copied back into the text in a different location. When text is sent out to the clipboard, its previous contents are overwritten. Text may either be copied from or cut out of the document. Later text may be "pasted" back in to the text in some other location.

The clipboard uses space on the text storage device (such as a floppy drive) that you selected when *Window Writer* was first started. If more than one file is being edited simultaneously and the *Window Writers* programs are using the same text buffer device, they will be able to share the same clipboard. Thus text can be transferred between different *Window Writer* programs that are simultaneously being run on the system. Remember, you also could accidentally overwrite text from one file with text from another.

MARK TEXT - <ALT><M>

When <ALT><M> is pressed, *Window Writer* will set a marker. You should set both a beginning and an ending marker. Once you set the second marker, the text that you marked will become highlighted. *Window Writer* will then expect you to enter one of the commands that deals with marked text such as cut, copy etc. If you type some other command, the markers will be removed and the text will no longer be highlighted. The mouse may be used to place the beginning and ending markers of the text to be highlighted. Place the mouse cursor at the

proper points, hit the mouse button, and the type the <ALT><M> control code.

CUT - <ALT><7>

This command cuts outs (deletes) the marked portion of the text and sends it out to the clipboard. Whatever was in the clipboard before is erased. The text may need to be reformatted. (Use <CTRL><A> to do this.)

COPY - <ALT><8>

This command copies the marked portion of the text and sends it out to the clipboard. No text is deleted from the document when this command is used.

PASTE - <ALT><9>

This command takes whatever is in the clipboard and "pastes" it into the document at the current cursor position. The text may need to be reformatted in order for it to appear as it should. (Use <CTRL><A> to accomplish this.)

SHOW - <ALT><0>

You may forget exactly what has been put on to the clipboard. This command allows you to view the contents of the clipboard before it is overwritten, pasted into the document, etc.

PARTIAL SAVE - <ALT><*>

This command copies the contents of the clipboard to a disk file so that a permanent copy may be made. This is how partial saves of the document are accomplished. If for example, the text is becoming too large and needs to be broken into smaller pieces, save as much as the clipboard will hold into different files. These files may be edited separately and later "chain printed" together to form a complete document.

7. SEARCH & REPLACE COMMANDS

There will often be instances when you will need to locate a word or phrase that is somewhere in a document. This is easily done using *Window Writer*'s global search and replace commands:

SEARCH - <ALT><S>

This command will prompt you for the "search phrase" and "replacement phrase" (if any) that *Window Writer* is to look for (and

replace if specified). You must also enter whether or not changes should be made in the entire document or not at all. If you enter 'y' for yes in answer to this query, *Window Writer* will automatically search and replace all occurrences of the specified sequence of characters that were entered. If you enter a 'n' at the query, *Window Writer* will simply search for the specified sequence of characters and reposition the cursor to that point. You may then manually replace the phrase by using the <ALT><R> command described below. If you wish to abandon the search while *Window Writer* is working you may do so by hitting <BREAK>.

It is also possible to include what are commonly called "wildcard" characters in the search phrase. Such wildcard characters are represented by a question mark "?". Wildcard characters are used to tell *Window Writer* that any character may be substituted into that position in the search phrase. For example, if *Window Writer* were told to look for "?arry", *Window Writer* would report words such as "marry" and "carry" as matches. Once the word or phrase has been found, you can replace it using the replacement phrase entered previously.

CONTROL CODE SEARCH - <ALT><J>

There is also a command to search for a control code within the text. You will be asked to select the code to search for from either the format code or style code menu. Upon selection, searching and replacing proceeds as it would with regular text.

MANUAL REPLACE - <ALT><R>

This command will manually replace the phrase searched for (if the phrase searched for was found) with the replacement phrase (if no replacement phrase was entered, then the search phrase will simply be deleted).

It should be pointed out that *Window Writer* will be able to search approximately six times faster if it is looking for a single word with no wildcard characters, rather than a phrase or wildcard characters. This is because it will not have to take into account the possibility that the search phrase is "broken" over two lines. (Only a phrase can be broken over two lines since the "word wrap" feature of *Window Writer* prevents single words from being broken over two lines).

FIND NEXT OCCURRENCE - <ALT><N>

This command will tell *Window Writer* to go look for the next occurrence of the search phrase. If the cursor does not move, it indicates that there are no more occurrences to be found in the remaining text.

8. TABULATION COMMANDS

As with most typewriters, *Window Writer* has a tab key, <F1>, that can be most helpful when columns are to be aligned or paragraphs are to start with an indent.

TAB RIGHT - <F1>

Hit this key in order to tab forward to the next tab stop. The cursor will not move if the cursor is already past the right most tab setting.

TAB LEFT - <SHIFT><F1>

Use this command to tab backwards to the previous tab stop.

SET TAB - <CTRL><T>

This is the command that is used in order to set a tab stop. Up to sixteen tab stops may be defined. In addition a customized set of tab stops may be defined and loaded into the computer automatically when *Window Writer* starts up. This is discussed in the section dealing with customizing the system.

CLEAR TAB - <CTRL><U>

This command is used to clear a tab stop that was previously set.

9. STYLE COMMANDS

UNDERLINING ON - <ALT><U>

This is the command to activate underlining. All text that is now typed will be underlined. Underlining is show on the screen display.

UNDERLINING OFF - <ALT><O>

This command will turn off the underlining feature. It may be necessary to refresh the screen in order to get a proper display of the text that is underlined. Do this by hitting <SHIFT><F2>.

ENTER STYLE CODE - <CTRL><F1>

This command will cause a style code menu to appear on the screen. The mouse should be used to select the desired code to be inserted. Then click the mouse button to select that code. More than one code can be inserted from the menu. Once all the codes have been selected, click the mouse on the "close" box in the upper left hand corner of the screen or hit <BREAK> to get back to the editor. A note should be made here of the way codes are handled.

Style codes are actually control codes (a special sequence of numbers) that affect the appearance of text on the screen. Style codes can cause

text to appear underlined or cause text to change color. They can also be completely invisible and have no apparent effect on the text on the screen. In order to see where exactly these codes are, use the <ALT><V> command. All style codes are stored in a table. When text is printed out to the printer, these style codes are converted into proper codes for the printer since the printer's style codes are not the same as the computer screen's. New style codes can be added or old ones modified easily. See "Other Utilities" for more information on this.

INSERT CONTROL CODE No. 2 - <ALT><.>

This command will insert at the current cursor position, the last control code that was previously entered. This is a quick time-saver command.

INSERT CONTROL CODE No. 1 - <ALT><,>

This command will insert at the current cursor position, the second to last control code (either format code or style code) that was previously inserted. This is a quick time-saver key command.

VISUALIZE LOCATION OF CODES - <ALT><V>

This will mark the position of all "control codes" on the screen with a blinking caret (^) symbol. In order to determine what the code actually is, position the cursor over the blinking caret and use the <ALT><D> command. To turn off this feature, type <ALT><V> again.

DISPLAY/DELETE CONTROL CODE - <ALT><D>

This will display the control codes at the current cursor location, one at a time and in plain English. It will also ask you if the code should be deleted. Note that control codes cannot be deleted using the normal character delete commands, <CTRL><D> and <CTRL><V>. This is to prevent their accidental erasure.

STRIP CONTROL CODES - <ALT><T>

Sometimes it will be necessary to remove all control codes in a file. Such a time would be when you want to run a spelling checker that cannot handle the control codes that *Window Writer* inserts into the text. Use <ALT><T> to remove all embedded control and formatting codes in the text. Such things as underlining and formatting commands will all be deleted, so be sure you save a copy of the text to disk first.

EXAMINE TEXT - <ALT><X>

This is one of the special, handy commands provided by *Window Writer*. It allows you to view the original text while it is being edited in another window. The first time that this command is invoked, only the lower half of the screen will remain accessible. The second time that this key is pressed, whatever was in the lower half of the screen is saved, and only the top half of the screen will become accessible. The third time the screen will return to normal.

HELP MENU - <ALT><H>

This command will activate the help menu selection screen. Four different help menus are available. Once activated the lower half of the screen will contain the "help", while the top half is used for editing. In order to turn off the help, type this command again and hit <ENTER>. Help menus are merely text files stored in the /dd/SYS directory. Feel free to customize them.

STAT UPDATE - <CTRL><O>

This is the general purpose "get stats" command provided by *Window Writer*. It will report such things as the number of words and lines in a file and the number of lines that there are unused in the text storage buffer.

SHELL - <ALT><C>

Use this command to invoke an OS/9 shell in an overlay window. All features regularly found in the shell will be available. Hit <CTRL><BREAK> to leave the shell.

RESTART - <CTRL><BREAK>

Use this command to start a new document. The text currently in the buffer will be erased so that a new unnamed file can be edited. Use the SAVE AS... command to name the document.

QUIT - <ALT><Q>

This command will exit the *Window Writer* program. Everything that was in memory will be lost, so be certain to save to disk the text that was being edited before you select quit.

REFRESH SCREEN - <SHIFT><F2>

This command will redisplay the screen. Multiple insertions and deletions of control codes sometimes leaves the screen in disarray. Use this function to get a correct display.

10. DISK FILE COMMANDS

FILE MENU - <ALT><F>

This command activates the file menu. All of the following file commands can be selected via a standard menu using this command. Several other commands are listed below which are selected from a menu reached by the <ALT><F> command.

LOAD FILE - <CTRL><K>

Use this feature to load in a file in the current directory from the disk drive. The directory listing of all text files on the disk will be displayed. Use the mouse to select the desired file. Note that this will erase the file currently in the buffer.

MERGE FILE - <ALT><K>

This command is similar to the previous command except that the text currently in the buffer is not erased. Instead, the new text is merged at the current cursor location with the text already in memory. Be sure to place the cursor where you want the merge to occur before you use this command.

SAVE FILE - <CTRL><C>

This command saves the contents of the buffer to the disk drive. If a file of the same name already exists on the drive, you will be asked whether it should be written over.

DELETE FILE - <CTRL><E>

Use this command to delete a file in the current directory. All files will be displayed in a menu just as for the load command. Use the mouse to select the file to be deleted.

RENAME FILE - <ALT><F>

This is the rename command. *Window Writer* will ask for the original file name and then the new name.

DIRECTORY LISTING - <ALT><F>

In order to get a complete listing of a directory, use this command. The 'Dir' utility supplied on the OS/9 Level II System diskette must either be in memory or in the current execution directory for this command to work correctly.

CHANGE CURRENT DIRECTORY - <ALT><Y>

To change the current directory, enter the new name at the prompt. *Window Writer* will not allow one to enter "pathlists" anyplace else except here. The file names given for the save command, for example, cannot contain a slash (/).

Chapter 3.

EDITING VERY LARGE DOCUMENTS

On occasion it may be necessary to edit files that are so big that the text storage device being used does not have the capacity to hold the entire document at one time. *Window Writer* has the ability to deal with such files. When a file is being loaded, *Window Writer* keeps track of its size. If the size exceeds 90% of the space available for text storage, *Window Writer* will stop loading in the file. Two temporary files will remain "open" or accessible from here on in. These files are created on the default drive called "/dd". The first file is called an "input file" and contains the text for which there was not enough room to load in. The other file is called an "output file" and is initially empty. You may retrieve lines from the input file by using the <ALT><B> command. However, room must be made for the lines to be retrieved. You must send some of the text to the output file using the <ALT><Z> command. Portions of the text are edited at any one time. After one portion has been edited, you must send it to the output file. This leaves room for some more lines to be retrieved from the input buffer. Eventually the entire document may be made to cycle through the editor. Once all changes have been made, use the standard file save command (<CTRL><C>) to save the text. If at any time you wish to go back to the top of the file, you may do so by first saving the file and then reloading it back in using the standard file load command.

GET LINES FROM INPUT FILE - <ALT><B>

This command gets lines from the input file if possible. You will be prompted to enter the number of lines to be retrieved. *Window Writer* will retrieve fewer lines if memory is full or the specified number of lines are not available from the input file.

ROLL LINES OUT TO OUTPUT FILE - <ALT><Z>

This command rolls out text to the output file so that more room is available in memory for more text from the input file. You will be prompted to enter the number of lines to be rolled out to the output file. *Window Writer* will send out fewer lines if the specified number of lines are not currently in memory or the default device /dd is full. If the device /dd is full, error number 248 - media full will be returned. Some unneeded files on device /dd must be removed in order to rectify the problem or use a second disk.

OPEN INPUT FILE - <ALT><F>

Use this command when you wish to append text to the end of a file that is currently being edited. Once this command is issued, the get lines and roll lines commands may be issued to get text (from the file that was specified) and send out text (to a temporary file) in the same way that *Window Writer* would when dealing with very large files.

OPEN OUTPUT FILE - <ALT><F>

Use this command when a file that is being edited becomes too large for the text storage device. *Window Writer* will report the error message number 207 - memory full when this happens. The roll lines command may be used to roll out text to the output file after this command is issued. This will make room in memory so that more text can be entered.

Chapter 4.

THE FORMATTER

Access the formatting menu by pressing <ALT><E>. Then you may use the arrow keys or mouse to highlight the selected option. Once the desired item is highlighted hit the <ENTER> key to perform the function. This section of the manual will cover the options displayed in the formatting menu.

1. FORM CONTROL UPPER MARGIN

Select this function to change the number of lines used for the upper margin. The upper margin must be large enough to accommodate a header if a header is being used.

LOWER MARGIN

Select this function to change the number of lines used for the lower margin. The lower margin must be large enough to accommodate a footer if a footer is being used.

LEFT MARGIN

Select this option to change the number of blank or space characters printed before the actual text line. Whenever possible, the printer's built-in left margin command should be used instead of *Window Writer*'s. Check your printer manual. This is to prevent conflict from arising when special features such as proportional printing and underlining are used. For example, some printers will actually underline the left margin that *Window Writer* prints out if an underline command is "broken" over two lines. To prevent such confusion as this, it is best to use the printer's own mechanism for setting the left margin.

LINE WIDTH

Select this function to change the number of characters printed per line. This value can be as large as 200 characters, which should be large enough to accommodate all print styles on most printers. Normal Pica-sized type is 10 characters per inch ans allows 79 characters per line.

LINES PER PAGE

Select this function to modify the total number of lines on a sheet of paper. This includes those lines used by the upper and lower margins. Thus if the upper margin and lower margin are both set at 5 and the lines/page is set at 66, the actual printed page will only be 56 lines in length.

SPACING

Select this function to change the spacing between lines. A spacing value of 1 indicates single spaced text. A value of 2 indicates double spacing. The value used must be less than the length of 1 page.

PAGE PAUSE

Select this function to tell *Window Writer* whether or not to pause after printing each page. Turn this feature ON if single sheets of paper are being used. Turn it OFF when continuous paper is available.

COLUMNS

A value other than 1 for this option indicates multi-column printing is desired. The line width should be set to the maximum width of the printed page divided by the number of columns desired minus two or three characters. In other words the line width should be set to the width of any one column. Thus if the total width of the page is 64 characters and two column printing is desired, the line width should be set to 30 and the columns option should be set to two. If a printer control code is not defined that will output reverse line feeds to the printer, then you will be instructed to manually reload the same sheet of paper every time a column is finished. You should position the paper to the first printed line, not the top of the page. The control code for a reverse line feed must be described as "REV". This is described in greater detail in the section discussing control codes in "Other Utilities."

PRINTER POSITION

Select this function when you want to start printing from some place other than at the top of the page. For example, if single sheet paper is being used and it is not possible to position the paper low enough for the print head to start printing at the top of the page, change this value to 3 or 4 instead of 1 to compensate for this slight shift. When using continuous fanfold paper, it will usually not be necessary to use this command.

JUSTIFY TEXT

Set this option to either left, right, center or full mode in order to determine how the text is to be justified. This feature may also be turned off if no justification is desired. Left justification aligns the text so that it is flush with the left margin and jagged on the right. Right justification aligns text so that it is flush with the right margin and jagged on the left. Center justification centers each line. Full justification "pads" the line with spaces so that both the left and right sides of the text are aligned with both margins.

HEADERS

A header is text that is printed out in the upper margin itself, such as titles and page numbers. In order to have a header printed out, it must already exist in a file (or it can be created using the Make-Header/Footer option described below). Simply enter the name of the header file and *Window Writer* will automatically print out that file at the top of every page. There are a few special symbols that can be used in a header file.

If a page number is to be included in the header, simply insert a '#' at the proper position in the line in which the page number is to be printed. If the page numbers will go to double digits, use '##'. If the page numbers will eventually reach triple digits, then use '###' in the header file. Thus if you have a 100-page document, place a line in the header file like this:

Page ###

When the computer actually prints out page 1, this part of the header will say:

Page 1

When the computer prints out page 99, the following will be the appearance:

Page 99

Finally when the computer gets to the last page it will print:

Page 100

The page numbers are right justified in this header field the correct number of digits.

It is easy to have different headers on even and odd pages of the print out. If a particular line in a header file is only to be printed on even numbered pages, then the last character on the line should be a '<'. If the line is only to be printed on odd numbered pages then the last character of the line should be a '>'. The number of lines in the header file should be less than or equal to the number of lines allocated to the upper margin.

FOOTERS

A footer operates in almost the same manner as a header except for the fact that it is printed out at the bottom of the page in the lower margin rather than at the top of the page. The number of lines in the footer file should be less than or equal to the number of lines allocated for the lower margin.

MAKE HEADER/FOOTER

This option allows you to create a header or footer file within the formatting module itself. You may, of course, use the *Window Writer* editor to create the header or the footer file instead. Please keep in mind that the left margin and line width settings have absolutely no control over how and where header/footer lines are printed. This allows printing left of the left margin and right of the right margin if desired. You will be prompted for the name of the file to be created (to hold the header or footer). You will not be able to write over a file that already exists so a name must be selected that has not already been used. The file will be created (in the /dd/SYS directory) automatically. Also you will be asked for the number of lines that the header or footer will be occupying. Following this, you will be able to enter the text for the actual header or footer. No editing features are available here except for backspace with delete.

PAGE NUMBER

This option selects what number is to be used for the first page number. The only way to have *Window Writer* print out a page number automatically is to include the page number symbol (##) in a header or footer file as described above.

MAIL MERGE

This function allows you to produce several copies of one document and have a personalized address on each one. The list of addresses must be contained in a separate file. Simply enter the file name containing the addresses and *Window Writer* will automatically access that list of addresses when necessary (you must enter the "Print Next Address" formatting code in the text at the location where the address is to be printed).

When printing the document, you must tell *Window Writer* the number of copies to be printed. For example, if you choose 5 copies to be printed and have entered a mail merge file name, then the first five addresses in the address list will be used on the five copies. Each individual address in the address file should be separated by a blank line. A typical address list file may appear as follows:

John Smith
2222 West Washington Street
Chicago, IL 50505

John Doe
3333 North Lincoln Avenue
Los Angeles, CA 90909

etc.

FILE NAME

This option selects the name of the file currently being printed. Usually this is the name of the file currently in memory but does not have to be. If the file name ends with a ".f" extension, *Window Writer* assumes that the text has already been formatted and saved to disk (see the "Output To" command). Thus *Window Writer* will not format the text at all, but print it out "as is". Print outs in this mode are usually faster because *Window Writer* does not have to spend time reformatting it.

ADJUST BAUD

This option lets you change the baud delay (speed) at which the printer operates. Presented here are the standard baud rates and their corresponding delay values:

BAUD RATE	DELAY VALUE
300	857
600	425
1200	209
2400	101
4800	47
9600	20
19200	6

Table 1. Printer Baud Rate Selection

OUTPUT TO

This option selects which device should receive the formatted output. The formatted document may be sent out to the screen (for previewing), disk, or the printer. If the screen is selected for output, an overlay window will cover the screen and the text will be displayed. Output may be paused by pressing the space bar. Also, the print operation may be terminated by pressing <BREAK>.

If a long document is to be printed or several copies are to be printed out, it may be helpful to first save the formatted output to disk and then print the formatted file. The formatted file will have an extension of '.f' added to it. When *Window Writer* prints out a file with a '.f' extension it knows that it is already formatted and does not spend time reformatting it, but prints directly from the disk.

PRINT

This option starts the printing process to the device selected as described above. You will be prompted for the number of copies to print.

SEND CODE DIRECTLY TO PRINTER

This option allows you to send control codes to the printer. Type a single number at a time followed by <ENTER>. Special printer commands such as near letter quality mode or left margin set may be selected here before printing a document.

3. DYNAMICALLY CHANGING FORMATTING CODES

One of the features of *Window Writer* that make it so versatile is its ability to change dynamically within the text various formatting options. You may enter a format command by pressing <CTRL><F2>. You may be asked for further information (a parameter) depending on the command you select. Following are the various formatting commands and a brief description:

TOP MARGIN

Top margin sets the number of lines for the top margin. This command takes effect at the top of the next page. The parameter value holds the number of lines to be used for the upper margin. The default is 5 lines.

BOTTOM MARGIN

Bottom margin sets the number of lines for the lower margin. This command takes effect at the end of the current page. The parameter value holds the number of lines to be used for the lower margin. The default is 5 lines.

LINES PER PAGE

Lines per page sets the number of lines (printed and unprinted combined) that a page holds. The parameter value holds the number of lines to be used for the upper margin, printed page and lower margin. The default is 66 lines. Enter this command on the first line of a page only.

NEW PAGE

New page command sets the printer to the top of the page before printing out the current line. The parameter given indicates whether multi-column printing is desired for the next page. The parameter specifies the number of columns to be printed. If multi-column printing is not desired, the parameter value should be set to one.

SPACING

Spacing sets the spacing between lines. A value of 1 in the parameter field indicates single spacing (the default). A '2' indicates double spacing etc.

LEFT MARGIN

Sets the left margin to the number of characters specified in the parameter field in the control code table. The printer's built in mechanism for setting the left margin should be used whenever possible and this setting should be left at zero. If the printer that is being used supports a left margin setting, use a standard control code at the beginning of each document to tell the printer to set the left margin to the desired value. (See your printer manual for the control codes.)

WIDTH OF LINE

Width sets the width of the printed line to the number of characters specified in the parameter field. The default is 70 characters.

JUSTIFICATION MODE

Justification mode changes the way text is justified according to the value in the parameter field as follows:

0) Justification Off

1) Full Justification

2) Center Justification

3) Right Justification

4) Left Justification

PROPORTIONAL SPACING

Proportional spacing allows you to use proportionally spaced characters when printing out to the printer. This command tells *Window Writer* that the text is to be proportionally spaced. It does not tell the printer. A proportional spacing style code should be entered in the text file also to tell the printer. Not all printers are capable of proportional print. Check your manual.

If the parameter field for this command is 0 then variable spacing is turned off; otherwise it is turned on. The parameter specifies what line number in the file 'spacing_filelist' contains the name of the proportional character set to load. See 'Modify the Proportional Character Set' for more on this. The following character sets are defined in your /dd/SYS directory.

pcs0 - standard draft mode 0

pcs1 - standard draft mode 1

pcs2 - Courier Near Letter Quality

pcs3 - Sanserif Near Letter Quality

pcs4 - Orator Near Letter Quality

NEW HEADER

You can change headers dynamically within the text by entering a 'New Header' format code. You will also be asked for the 'Line Number in File.' The number you enter specifies the line number in the file '/dd/SYS/header_filelist' that contains the name of the new header file. For example, if you wish in the middle of your document to start numbering pages, you may do so by entering a 'New Header' format code while in the editor. When you are asked for the 'Line Number in

File' enter the number '1.' Save the current document and then start a new one (press <CTRL><BREAK>).

You must first create the header file. Type the following:

Page ##

Now issue a change current directory command and change the current directory to '/dd/SYS.' Save this file with the name 'header1.' Next you must edit the 'header_filelist' file. Open this file for editing. On the FIRST LINE of the 'header_filelist' file type:

```
header1
```

That is all that is necessary. Save this file. Then change the current directory back to what it was previously (usually /dd/TEXTFILES). Now when *Window Writer* comes across the 'New Header' command it will look at the first line of the file 'header_filelist.' The first line of this file contains the name 'header1.' *Window Writer* will open the file 'header1' and print it at the top of the next page and following pages.

NEW FOOTER

New footer command allows one to change the footer in the middle of a document. The syntax for this command is the same as for the NEW HEADER formatting command. Follow the same Save/Create procedure outlined above.

PAGE NUMBER

Page number sets the current page number to whatever number is specified in the parameter field. The result of this change will not be visible unless a header or footer file is being used that prints out page numbers.

CHAIN PRINT

Chain file command allows you to edit a large document one section at a time and when printing have them all combined into one document. This command should appear at the end of each file that is to chain another. In order for this command to work properly, the files being chained must be properly named. The name of all the files to be chained printed must be exactly the same except for the last one or two characters at the end. The first file's name in the chain should end with a '..1', the second file to be chained should end with a '..2' and so on. Thus, a group of files could be named 'Letter..1', 'Letter..2', and 'Letter..3' or perhaps 'Chapter..1', 'Chapter..2', and 'Chapter..3'.

MERGE PRINT

This merges another document into the middle of the one currently being printed. When *Window Writer* encounters this command, it will

open up the file specified and start printing it. When it is finished printing the new file it will continue with printing the old file where it had left off. The format for the parameter is the same as that described in the NEW HEADER command. The number you specify indicates the line in a file called 'merge_filelist' that contains the name of the file to be merged.

This command can also be nested. What this means is that you may merge a file, and that file in turn may call this same formatting command and merge in another file and so on. At some point, OS/9 may become overloaded and report an error. This is to be expected and cannot be avoided. Try to use the CHAIN command instead of this one as that command does not have this limitation.

PRINT NEXT ADDRESS

Gets the next address from an address list that is in a file specified under the 'Mail Merge Name' option in the formatting menu. When Window Writer encounters this command it will try to read an address from the Address List file and print it out at the current printer position. The first word and last word on the first line of the current address is stored in the computer's memory. Usually these two words will be the first and last name of the person being addressed. As described below, one may then tell *Window Writer* to print these words at certain times so that a personalized letter can be made quickly and easily.

PRINT FIRST NAME

First name of the person currently being addressed is printed out at the printer's current position.

PRINT LAST NAME

Last name of the person currently being addressed is printed out at the printer's current position.

PRINT DATE

The current date that was set when the system was first booted will be printed out at the current cursor position. Use the SETIME utility (provided on the OS/9 System Disk) in the startup file to set the date if it is incorrect.

(DIS)ABLE FORMATTING

Turns the alignment feature on or off depending on the parameter field. If the parameter number is 0 then alignment is turned off; otherwise it is turned on. The alignment feature should be turned off whenever

formatting is not desired, such as when tables or charts are being printed.

Also, justification is disabled for the block of text not to be formatted. Justification (if it had been set to some mode before) will resume when formatting is turned back on.

Chapter 5.

OTHER UTILITIES

Included in the *Window Writer* package are three utilities that will help to increase the ease of use and improve the flexibility of the system. All three of these utilities are accessible from the formatting module of *Window Writer*. The first utility is the control codes modifier.

1. MODIFYING THE STYLE CODE TABLE

When you want to insert special codes (such as underlining) into the text, it is a fairly trivial matter. Simply type <ALT><U> to start underlining or <CTRL><F1> to access the style code table, move the cursor over the 'Underlining On' option and hit the button. *Window Writer* is actually inserting a code into the text. This code is called an escape code or a control code. It tells the computer to turn underlining on. A conflict arises when the material is to be printed out however. The control code to turn underlining on is different for the printer than it is for the computer. Thus the computer control code to turn underlining on must be "translated" into a code that the printer understands. This is done via the Control Code Table. This table can contain up to 63 "conversions" or entries for control codes.

Each entry is given an English description (no longer than 25 characters) so that you can understand what that code stands for. Also included with the description is the screen control code and the corresponding printer control code. When text is displayed on the screen, the screen control code is used. When text is printed out to the printer, the analogous printer code is used. A screen code may be up to 5 characters long. The printer code may be up to 10 characters long. Please note that the screen code need not perform the same function as a printer control code. For example, one may define red-colored text on the screen as double width text on the printer. Also there need not be a corresponding printer code for every screen code. This is how some formatting commands are handled. For example, one may enter an invisible screen code that is used to represent a formatting command such as resetting the line width. This code tells *Window Writer* that a new width is desired when printing out. However, the printer does not

need any information from this screen code and so there is no corresponding printer code.

NOTE: each entry in the table must be represented by a different screen code. In order to define an entry into the control code table, one must use the "embedded control code modifier" utility.

There are six options available in the control code modifier utility:

DISPLAY ENTRY

This will allow you to look at the screen codes and the printer codes for a previously defined entry. Note that no two screen style codes in the control code table should be the same.

ADD ENTRY

Select this item to enter a new code into the menu table. A unique invisible code sequence will be displayed that you can use as the screen code. This screen code will not affect the color of the text. If you wish to enter that type of code, read the section on technical information below. Next, you will be prompted for the analogous printer codes. These codes can be found in the user's manual that came with your printer. If there is no analogous printer code, simply hit < ENTER >. Last, you should enter a (maximum) 25 character English description (anything you want) so that you can later reference this entry. If you do not add a description, the entry is not protected from overwriting. (See below).

DELETE ENTRY

Select this item to delete an entry. All this command does is wipe out the English description of the entry you specify. When @BOLD COMMANDS = ADD ENTRY is later invoked it will see this entry as being free and allow it to be completely overwritten with new data.

MODIFY ENTRY

This item will allow you to modify parts of an entry. If you do not wish to change a certain part simply hit < ENTER > and nothing will be changed. To change something simply enter a new value.

LOAD MENU TABLE

This will allow you to load a control code table other than the one normally used (called 'esc_table'). This may be used when switching to a different printer for example. Menu tables for several common printer types are included in this file including: IBM, Epson, Okidata, and Tandy.

SAVE CURRENT TABLE

This option allows you to save the current control code table after making modifications to it. The default file name is 'esc_table' and is the table that is initially loaded when *Window Writer* first starts up. If a file name that already exists is specified, the old file will be overwritten by the new one.

QUIT

In order to quit, click the mouse on the "close window" box in the upper left hand corner of the window. You can also press <BREAK> to return the main formatting menu.

2. TECHNICAL INFORMATION ON STYLE CODES

Style codes are stored in a menu table. This menu table is actually a disk file that contains style codes, their analogous printer codes, and an English description of what each entry does. There are essentially three types of style codes: invisible, those that color the text that follows, and visible single character codes. When text is printed out, the style code for the screen is replaced with the analogous printer code.

Effects of Code	Sequence of Codes to Enter
Text becomes green (color 3)	27 50 51
Text turns back to white	27 50 82
Text becomes blue (color 4)	27 50 52
Text turns back to white	27 50 74
Text becomes red (color 5)	27 50 53
Text turns back to white	27 50 66
Text becomes pink (color 6)	27 50 54
Text turns back to white	27 50 58
Text becomes yellow (color 6)	27 50 55
Text turns back to white	27 50 50

Table 2. Possible Foreground Style Codes

Invisible style codes consist of the sequence of numbers 27 63 zz; where zz is a unique number in the table.

There are two types of style codes that color text. One type colors the foreground (the color of the characters) and the other type colors the background (the area behind each character). They work in a similar

manner. **Remember that no two entries can ever have the same style code.**

A conflict arises here because of this. For example suppose you would like to define two new style codes. One to represent bold faced type and the other to represent italics. The bold faced type will be represented by the color red while the italic type will be represented by the color blue. Codes must also be defined to turn off the bold faced type as well as the italic type. The codes for both of these must turn the text on the screen back to the standard white color yet both style codes must be different. Luckily more than one style code exists to turn text white. The above Table 2. summarizes what foreground style codes should be used.

Please note that the above codes deal with colors 3 through 7. Colors 0, 1, and 2 are used by the system and should not be used in style codes. In order to use the background style command, use the codes shown in Table 3.

Effects of Code	Sequence of Codes to Enter
Background becomes green (color 3)	27 51 51
Background turns back to black	27 51 80
Background becomes blue (color 4)	27 51 52
Background turns back to black	27 51 72
Background becomes red (color 5)	27 51 53
Background turns back to black	27 51 64
Background becomes pink (color 6)	27 51 54
Background turns back to black	27 51 56
Background becomes yellow (color 6)	27 51 55
Background turns back to black	27 51 48

Table 3. Possible Background Style Codes

Caution: Do not set the foreground and background to the same color, or your text will be invisible.

The last type of style code is the single character visible code. This type of code includes special symbols and foreign characters that are outside the standard character set. These codes may be inserted by first pressing <F2> followed by another key. The ASCII value of this key will have 128 added to and it will be inserted into the text. In order to have your printer print out these characters correctly, they must be defined in the style codes menu table. Be sure to enter the corresponding printer code for the character to be generated. The printer code for various characters can be found in your printer's users manual.

EXAMPLE 1:

Suppose that you wish to add an entry into the control code table that would turn on bold faced printing. If you have decided to represent boldface text as bright red text (this is assuming something else is not represented by red text):

1. Once in the formatting module (by hitting <ALT><E>), you should move over the 'Modify Control Code Table' option and hit <ENTER>.

2. Then select the 'Add Entry' option.

3. You will be asked to enter the screen codes. The screen code for red text (listed above) is 27, 50, 53. When asked to enter screen code no. 1, enter the number 27 and hit <ENTER>. When asked to enter screen code no. 2, enter the number 50. Do the same for the number 53. When asked for the fourth screen code, just hit <ENTER> as you have no more numbers to enter for the screen code.

4. Next you will be asked to enter the printer codes that correspond to the screen codes just entered. On IBM compatible printers, the codes to turn on bold faced type is 27, 71. Enter the numbers in sequence just as you did above.

5. Lastly you will be asked for a description of this entry. Enter something such as 'Bold Face ON'.

6. This code is now defined. You may verify this by using the DISPLAY ENTRY option. Press <BREAK> to return to the main formatting menu.

EXAMPLE 2:

Now suppose that you wish to add an entry into the control code table that would turn off bold faced printing. On the screen, this code will turn the text back to its normal color of white.

1. Once in the formatting module (by hitting <ALT><E>), you should move over the 'Modify Control Code Table' option and hit <ENTER>.

2. Then select the 'Add Entry' option.

3. You will be asked to enter the screen codes. The screen code to cancel red text (listed above) is 27, 50, 66. When asked to enter screen code no. 1, enter the number 27 and hit <ENTER>. When asked to enter screen code no. 2, enter the number 50. Do the same for the number 66. When asked for the fourth screen code, just hit <ENTER> as you have no more numbers to enter for the screen code.

4. Next you will be asked to enter the printer codes that correspond to the screen codes just entered. On IBM compatible printers, the codes to turn off bold faced type is 27, 72. Enter the numbers in sequence just as you did above.

5. Lastly you will be asked for a description of this entry. Enter something such as 'Bold Face OFF'.

6. This code is now defined. You may verify this by using the DISPLAY ENTRY option. Press <BREAK> to return to the main formatting menu.

3. THE CURRENT STYLE CODE TABLE

Four different style code tables are supplied with the *Window Writer* system. They are located in the SYS directory and are called 'ibm_table', 'epson_table', 'oki_table', and 'tandy_table'. *Window Writer* uses a file called 'esc_table' - this is simply a copy of one of the other tables.

The following codes have already been defined and are compatible with most IBM, Epson, Okidata, and Tandy printers. If you would like to change the appearance of any code on the screen, change it as described above.

UNDERLINING ON

Turns on underlining. The screen will display text that is underlined when this code is inserted into the document.

UNDERLINING OFF

Turns off underlining. The screen will stop display of underlined text.

BOLD FACE ON

Selects bold face type on the printer. Represented by foreground color number 7 (yellow) on the screen.

BOLD FACE OFF

Deselects bold face type on the printer. Represented by the standard color for text, color 5 (white).

EMPHASIZED ON

Selects emphasized printing on the printer. Represented by foreground color 6 (pink).

EMPHASIZED OFF

Deselects emphasized printing on the printer. Represented by the standard color for text, color 5 (white).

ITALICS ON

Turns on italics on the printer. Represented by the foreground color number 5 (red).

ITALICS OFF

Turns off italics on the printer. Represented by the standard color for text, color 5 (white).

SUPERSCRIPTS ON

Selects superscripts on the printer. Represented by the foreground color number 4, (blue).

SUBSCRIPTS ON

Selects the subscripts on the printer. Represented by the foreground color number 3, (green).

SCRIPTS OFF

Turns off both superscripts and subscripts on the printer. Represented by the standard color for text, color 5 (white).

NORMAL SIZE TEXT - 10 CHARACTERS PER INCH

Restores the normal Pica type. Represented by an invisible code.

ELITE SIZE TEXT - 12 CHARACTERS PER INCH

Turns on the Elite mode. Represented by an invisible code.

CONDENSED TEXT - 17 CHARACTERS PER INCH

Turns on the condensed mode for the printer. Represented by an invisible code.

ENLARGED TEXT - 1 LINE AT 5 CHARACTERS PER INCH

Turns on double width text for the remainder of the current line. Represented by an invisible code.

PROPORTIONAL SPACING ON

This invisible code tells your printer to start printing in proportional type. This code should be accompanied with the proportional spacing format code (to tell *Window Writer* as well that text is to be proportionally spaced).

PROPORTIONAL SPACING OFF

Turns off proportional spacing on your printer. This code should be accompanied with the proportional spacing format command to tell Window Writer to stop formatting text with proportional spacing.

FILLER CHARACTER

This character is a special character (degree symbol) that has no analogous printer code. It should be used after every character that follows the ENLARGED TEXT command. This is *Window Writer's* way to indicate text which is twice as large as normal text. Such things as centering and justification will then all work properly.

CONCATENATION CHARACTER

This character is used to represent the concatenation of two lines and is used when a file is read in that contains a line that is longer than 80 characters in length. When the file is resaved to disk, the lines are rejoined together and the concatenation character is removed. This character is represented by the ASCII code of 173. It can be generated by pressing <F2><->.

LINE FEED CHARACTER

ASCII character number 10. Use in code search & replace to remove line feeds from a BASIC09 program so that it can be edited properly.

SPACE CHARACTER

ASCII character number 32. Same as character produced when hitting the spacebar. Use this code to replace the line feeds that are in a BASIC09 program.

VARIABLE SPACING CODE

The printer code sequence that is used to change the spacing between characters in amounts as small as 1/240 of an inch. This code is used by *Window Writer* to fully justify proportional text. Full justification of proportional text cannot be achieved if this code is not supported by the printer.

HORIZONTAL TAB CODE

The printer code sequence that is used to move the print head of the printer to the right in increments as small as 1/120 of an inch. This code is used by *Window Writer* to right and center justify proportional text. Right and centered justified proportional printing is not available if this code is not supported by the printer.

REVERSE LINE FEED CODE

The printer code sequence that is used to generate a reverse or backwards linefeed. Paper should move down instead of up when this command is given. *Window Writer* uses this printer code to automatically reposition the printer at the top of the page when printing in a multi-column format. If your printer does not have this feature, then this entry should be deleted from the Control Code Table. *Window Writer* will then manually instruct you to position the paper to the top of the page during multi-column printing.

4. PROPORTIONAL CHARACTER SETS

When you want to print out text in proportional spacing (that is, each character printed takes up a different amount of space) the computer must calculate the width of each line by adding up the number of spaces each character takes so that it knows where the line must be split. A table is used to hold the width of each character. *Window Writer* adds up these widths. A few standard tables are provided that should be compatible with the proportional character sets of most printers. If other sets are desired, they may be created if you know (from the instruction manual for the printer) the number of dots horizontally that each character takes up. If you do not know, there is a way to find out.

If your character set is not properly included in the existing menus, it is necessary to determine the number of dots that a character occupies when printed. This is a fairly easy, though somewhat tedious process. To simplify determination, you may use a small BASIC program under RSDOS such as this:

```
10 INPUT "CHARACTER: ",A$
20 PRINT "HIT Q WHEN ONE COM-
PLETE LINE PRINTED"
30 PRINT "KEEP HITTING ANY
OTHER KEY OTHERWISE"
35 X=0
```

```
40 B$=INKEY$:IF B$="" THEN GOTO 40
50 IF B$="Q" OR B$="q" THEN GOTO 80
60 PRINT #-2,A$;
65 X=X+1
70 GOTO 40
80 X=INT(1920/X)
90 PRINT #-2, X; "DOTS USED BY THE
"; A$; " CHARACTER"
100 GOTO 10
```

Before using this program, put the printer in the proportional mode that is to be used (this should be covered in the printer's manual). Type in and run the program above. Enter a character when asked to. This will be the character for which the computer will determine the number of dots occupied. Keep pressing any key until the printer is forced to go to the next line. This tells the program the number of characters that will fit on one line. From this information the computer will report the number of dots that one character occupies. Write this down. Repeat the process for all the ASCII characters including punctuation and the space character. Once this data has been obtained, you are ready to enter it into *Window Writer*.

First the file name for the new proportional character set should be entered. If the file you name already exists, it will be read into memory so that it may be modified. If the file did not exist before, it will be created and will not contain any meaningful information.

Secondly, you should use the arrow keys to move through the characters that are to be defined. On the left the "ASCII value" for the character is displayed and on the right the character itself, followed by the number of dots each character takes up.

Hit <ENTER> when a width value is to be changed. At the prompt enter the number of dots that character occupies when printed. Hit <ENTER> once more when done.

Repeat this process until all the widths have been entered. Then press the 'Q' key to quit modifying the table. You will then be returned to the formatting menu.

APPENDIX I

CUSTOMIZING THE SYSTEM

Although the default settings of *Window Writer* were designed to handle most standard printing needs, its default settings and features can be customized. There are a number of items that can be changed. This section of the manual will go over these items.

1. THE ENVIRONMENT FILE

The secret to this versatility lies in the fact that *Window Writer* calls upon what is called an environment file during initialization. This file contains a variety of information, some of which *Window Writer* does not use. (The other information is present to maintain compatibility with the Multi-Vue software package.) The default contents of the environment file along with a brief description of each item are listed here. (Items beginning with asterisks are comments):

RBFDEV = /d0,/r0 Devices that can be used to store text such as floppy drive, RAM drives, etc. Here are some conventions for device names:

/d0 - a floppy disk device

/r0 - a RAM disk device

/h0 - a hard disk device

SCFDEV = /p,/t1 Not Used

MONTYPE = 1 Determines the monitor type:

0 - Composite

1 - RGB

2 - Monochrome

*RAM = 128 Not Used

RAM = 512

DATA = /d0/TEXTFILES This option allows one to have a default data directory available whenever Window Writer starts up.

EXEC = /d0/CMDS

*PROGRAM = Shell

*PARAM = i = /term

REPSTR = 2 Starting delay value for key repeat:

1 - no key repeat

2 - long delay

3 - medium delay

4 - short delay

REPSPD = 2 Repeat key speed:

1 - slow through

5 - fast

PTRRES = 1 Resolution for mouse:

0 - low resolution (64x64)

1 - high resolution (640x192)

PTRSID = 1 Determines port for mouse/joy-stick:

0 - left side

1 - right side

LFTMRGN = 5 Left margin default value (in characters).

LNLEN = 70 Line width default value (in characters).

SPACING = 1 Line spacing (1 indicates single spacing).

JUSTMODE = 0 Justification Mode:

0 - justification off

1 - full justification

2 - center justification

3 - right justification

4 - left justification

PGWDTH = 80 Not Used

HDRSIZ = 5 Upper margin value (in lines).

TRLSIZ = 5 Lower margin value (in lines).

PGLEN = 66 Page length (includes upper and lower margins).

TABSIZ = 4 Not Used

PGPAUS = 0 Pause at page bottom (0 indicates no pause).

PRPORT = /p Printer Device Name.

STDOUT = 1 Standard print device to be used by formatter:

0 - Screen

1 - Printer

2 - Disk file

PRNAME = DMP100 Not Used

PBDLY = 20 Printer Baud Rate Delay Value. (This value is for 9600 baud, see Table 1.)

(Palette Values with 0-7 background and 8-15 foreground)

PALET00 = 0,0,0 Color of background (black)

PALET01 = 1,1,1 Color of border (dark gray)

PALET02 = 3,3,3 Color of text and cursor (white)

PALET03 = 0,1,0 (dark green)

PALET04 = 0,0,2 (dark blue)

PALET05 = 1,0,0 (dark red)

PALET06 = 1,0,1 (dark pink)

PALET07 = 1,1,0 (dark yellow)

PALET08 = 0,0,0 Color of text when highlighted (black)

PALET09 = 2,2,2 Color of text in status window (light gray)

PALET10 = 3,3,3 Color of background when text highlighted (white)

PALET11 = 1,3,1 (bright green)

PALET12 = 1,2,3 (bright blue)

PALET13 = 3,0,0 (bright red)

PALET14 = 3,1,3 (bright pink)

PALET15 = 3,3,1 (bright yellow)

TAB0..TAB15 Tab default settings (255 indicates unused setting)

2. THE RAM DISK

Window Writer requires that a RAM disk (not necessarily its own RAM disk, but just some device called /r0) be present before it will operate. It uses the RAM disk for temporary storage of its files and also for some System Files such as the control code table and function key table. *Window Writer* will also use the RAM disk for text storage if it is selected as the storage device. The RAM disk that is supplied with the *Window Writer* package behaves exactly as though it were a temporary floppy drive (as described in the Getting Started section). Naturally, the RAM disk takes up memory space (unlike a floppy drive) that could otherwise be used for other programs that are running at the same time the RAM disk is in use. The startup file sets up the RAM disk for operation.

You may examine the startup file, but a brief overview of its operation will be explained here.

The RAM disk must first be loaded into memory, and it actually consists of three separate programs. All three files can be loaded at once to conserve memory by issuing a LOAD RCDisk command (RCDisk has all three programs merged within it). The first program is what is called a "device driver." This is the program that actually communicates with the RAM disk. The second program is called a "device descriptor" which holds a few default settings (initialization values) that the device driver uses. The third and last program is called "RDForm", and as the name implies it formats the RAM disk. After all three programs are loaded, they must be set up. First type 'iniz /r0' then issue the RDForm command. RDForm will expect an integer input on the command line followed by < ENTER > such as: RDForm 20.

The integer value determines the number of contiguous blocks of memory (tracks) the RAM disk should be allocated. Each block of memory is equal to 8,192 bytes of memory. If there is not enough memory available, an error will be reported. Once memory is allocated, the RAM disk must be formatted. RDForm does this automatically. Note that RDForm cannot be used with floppy drives or hard drives. The RAM disk is now ready to use.

If one wishes to remove the RAM disk (not while *Window Writer* is active), one must tell RDForm to allocate 0 blocks of memory for it. The memory that the RAM disk normally uses will be freed. One may also reallocate the RAM disk to a different size if desired simply by giving another RDForm command. Remember that RDForm formats the RAM disk and thus the RAM disk loses all of its contents each time RDForm is called!

If you wish to change the name of the RAM disk from /r0 to something else (remember *Window Writer* requires that some device called /r0 be present), you may do so by using the ModPatch utility supplied on the OS/9 system Disk. In the device descriptor "r0" at offset $21, the name begins. Note that the last character of the name has the high bit set (has 128 added to it ASCII value). Also the name of the RAM disk must be changed in the RAM disk formatter, RDForm. In the RDForm module at offset $13, the name begins. The last character of the name does not have the high bit set. Also note that the length of the name cannot be changed.

APPENDIX II

TROUBLE SHOOTING

ERROR MESSAGES

UNKNOWN PROCEDURE - ERROR No. 43

This error would occur if all the necessary *Window Writer* subprograms had not been correctly loaded into memory.

Check to see that all the instructions were followed when creating a working copy of the *Window Writer* disk.

ILLEGAL ARGUMENT - ERROR No. 67

This error will occur if you type an invalid value when asked to enter something. For example if *Window Writer* expects a number, but instead you enter a letter, this error will occur.

Simply reenter the correct value.

TAB TABLE FULL - ERROR No. 128

This error would occur if 16 tab stops are already defined and you try to define another one. *Window Writer* limits the number of tab stops to 16.

Clear some previously defined tab stops to remedy the situation.

LINE FULL - ERROR No. 129

The current line cannot hold any more characters. Each line can hold up to a maximum of 100 characters including invisible control codes.

To overcome this limitation delete some control codes or split some of the control codes over two lines if possible. Note that a short line will automatically be joined together with the following line when printed. The 100 character limit does not apply when sending text to the printer.

CLIPBOARD FULL - ERROR No. 130

There is not enough room on the RAM disk for more text to be placed on the clipboard.

More room can be made available for the clipboard by increasing the size of the RAM disk when it is initially created in the "startup" file.

MEMORY FULL FROM CLIPBOARD TRANSFER - ERROR No. 131

An attempt was made to transfer text from the clipboard that resulted in a memory full error.

There are several solutions to alleviate this problem. They are outlined in solutions to MEMORY FULL - ERROR No. 207 discussed later.

PATH TABLE FULL - ERROR No. 200

This error would occur when attempting to use the "Merge" formatting option one too many times within the same document. This is a system limitation and cannot be easily circumvented.

Try to use the "Chain" formatting option instead of this one whenever possible.

MEMORY FULL - ERROR No. 207

This error would occur when the text storage buffer (created when *Window Writer* was started) becomes full.

There are basically two solutions to this problem. The first is to save the text that is currently being edited. Then quit from Window Writer and start up again by typing 'ww' at the OS/9 prompt. When asked for the text buffer size, enter a larger value than that what was entered before.

The second solution is to open up a disk file (on device "/dd") and use it to store some of the information. Use the "Open Output File" option for this.

NO PERMISSION - ERROR No. 214

This error would occur if *Window Writer* tried to read or write to a directory rather than to a file. Such access is not permitted by the system. Also this error would occur if a file has its attributes set or reset such that it cannot be read or written to at all.

The solution to this error is to use the "attr" utility supplied on the OS/9 system disk. See the OS/9 manual for more information on this.

BAD PATH NAME - ERROR No. 215

This error would occur if an attempt is made to save a file under an illegal name. A legal file name must begin with a letter and can be composed of letters, numbers, the underscore character (_) and the period (.).

Use a legal file name to solve this error.

PATH NAME NOT FOUND - ERROR No. 216

This error would occur when a file or directory name was entered that does not exist.

Use the 'List Dir.' option to see what files exist.

MODULE NOT FOUND - ERROR No. 221

This error occurs when a needed module is not available in memory. For example suppose a request to change the current directory to '/d7' was issued. Such a device does not exist and ERROR 221 would be reported.

SYSTEM RAM FULL - ERROR No. 237

This error occurs when the system's own 64K space becomes full.

Make sure that all device drivers and device descriptors are in the bootstrap file. Also try reducing the number of non-essential program modules from the Bootstrap file. Refer to the OS/9 manual for more details on accomplishing this.

DEVICE NOT READY - ERROR No. 246

This error would occur if *Window Writer* is given the command to send text to the printer, but the printer is OFF LINE.

Turn the printer ON LINE to proceed.

MEDIA FULL - ERROR No. 248

This error occurs when an RBF type device (e.g. floppy disk, RAM disk) becomes full.

Delete some unneeded files from the device that is full to remedy the situation.

RECORD LOCKED OUT - ERROR No. 252

This error will not always be reported. This "locked out" condition occurs when two processes (e.g. two copies of Window Writer simul-

taneously running) both try to access the same disk file at the same time. One of the processes will be "locked up" and will appear to have crashed. This is what will happen if both processes are accessing the "esc_table" file that is used to store the control codes for example.

The process that is not locked up should finish whatever it is doing and release the file for other processes to use.

NON-SHAREABLE FILE BUSY - ERROR No. 253

This error occurs in a situation such as when one process tries reading from a file while another is trying to delete the same file.

Do not try to delete a file when another process is using it.

Index

J

L

M

P

Q

R

S

T

U