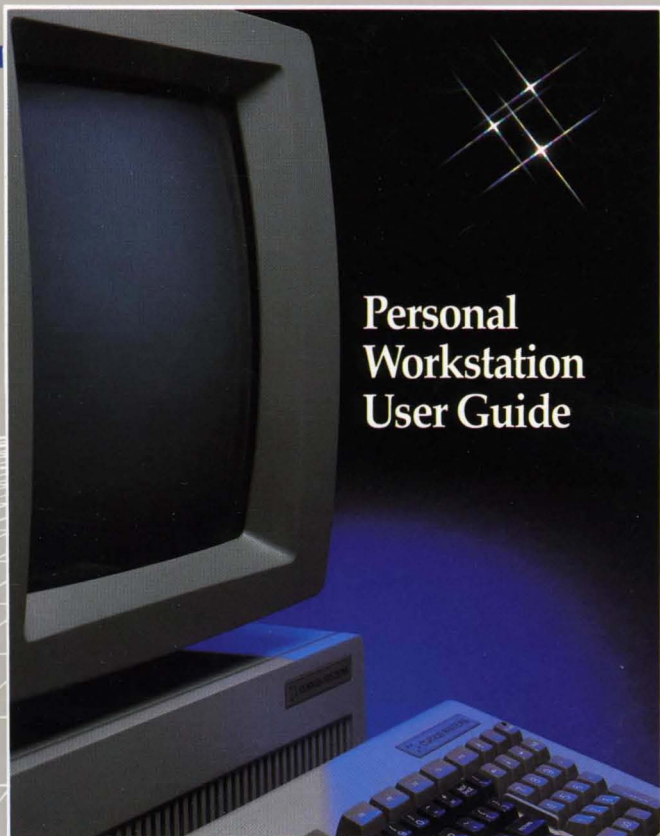


# THE CORVUS CONCEPT



Personal  
Workstation  
User Guide

  
CORVUS

# The Corvus Concept Personal Workstation User Guide

**Part No.:** 7100-02826

**Document No.:** CCC/30-22/1.4

**Release Date:** September, 1983

CORVUS CONCEPT™ is a trademark of Corvus Systems, Inc.



# TABLE OF CONTENTS

SCOPE	1
<b>CHAPTER ONE: STARTING OUT</b>	<b>7</b>
Power-on	7
Self-diagnostic	8
Log-on	10
Setting the System Date	13
Setting the System Time	15
<b>CHAPTER TWO: SYSTEM OVERVIEW</b>	<b>19</b>
The Keyboard	19
Function Keys	20
Function Key Labels	21
Modifier Keys	23
Cursor Keys	23
The Video Display Unit	24
Interrupting Screen Display	24
Screen Configuration	24
System window	25
Command window	28
Label window	28
The Dispatcher	29
How to Run a Program	32
The Restart Function	33
The Help Label	34
The Exit Label	35
Rebooting the System	35
Warm Boot	36
Cold Boot	36

<b>CHAPTER THREE: MANAGING FILES</b>	<b>37</b>
Volumes	37
Logical Units	38
The System Volume	38
The Current Volume	38
Setting the Current Volume	39
Listing a Volume	40
Slash /	44
Exclamation Point !	46
Managing Files	48
The File Manager	49
File Names	52
Wild Card Characters	53
Deleting a File	53
The Q Option	55
Deleting a Temporary File	56
Recovering Lost Files	57
Copying a File	60
The D Option	62
The S Option	62
Concatenating Files	63
Renaming a File	65
Listing a File	66
Redirecting Output	67
Printing a File	68
Editing a File	69
Loading the 7 by 11 Character Set	69
Creating an EdWord Workspace	70
Loading a File	73
Saving a File	74
Exiting EdWord	75
Re-entering the Workspace	75

Changing a Volume	76
Reclaiming Volume Space	76
<b>CHAPTER FOUR: MANAGING WINDOWS</b>	<b>79</b>
Window Management	79
The Window Manager	80
Clear Window	82
Reverse Background	82
Scroll Mode	83
Displaying Test Patterns	84
Creating a Window	85
Selecting a Window	87
Deleting a Window	88
Creating a Temporary Window	88
Print Window	89
<b>CHAPTER FIVE: CHARACTER SETS</b>	<b>91</b>
Character Sets	91
Loading a Character Set	92
Loading a Display Character Set	92
Loading a Keyboard Character Set	93
Alternate Character Sets	94
The 9 by 14 Alt Character Set Table	95
Editing a Display Character Set	98
The Edit Mode	104
Saving a Character Set File	112
Using the Test mode	113
Exiting the Character Set Editor	115

<b>CHAPTER SIX: DISK UTILITIES</b>	117
Spooling and Despooling	117
Spooling a File	118
Spool Options	119
Spool	123
Despooling a File	124
Volume Managmeent	127
Display Volumes Accessible	130
Display Volumes Mounted	131
Selecting a Drive	132
Changing Volume Attributes	133
Exiting the Mount Manager	135
The Volume Utilities Program	135
Backing Up Volumes	137
Restoring a Volume	140
Copying Diskettes	142
Setting the Floppy Unit Number	144
Setting the Work File	144
Reading a Diskette	145
Writing a Diskette	146
Deleting a Work File	147
System Status Information	148
<b>CHAPTER SEVEN: ADVANCED FEATURES</b>	149
Using Exec Files	149
Automatic Startup File	151
Defining Userkeys	153
Using the Runtime UCSD p-System	157
Setting Data Communication Parameters	158
Assigning a Driver	167
List Drivers On-line	169
Running the Terminal Program	171

The Floppy Utility Program	176
The Floppy Formatter	177
Formatter Options	177
Formatting a Floppy	181
Writing the Directory	183
Configuring the Floppy Driver	184
Writing the Boot Code	187
<b>CHAPTER EIGHT: PERIPHERAL CONNECTIONS</b>	
Type of Boot	189
Diskette Connection	191
Local Disk Drive Connection	191
Omninet Connection	192
RS-232C Ports	193
Connecting a Local Printer	193
RS-232C Interface Specification	196
Overview	197
Line Assignments	198
Protocols	201
Handshaking Protocols	202
Baud Rate	203
Parity and Character Size	204
Software Handshaking Protocols	205
Line Handshaking Protocols	206
Null Modem Connector	207
Changing the Screen Orientation	208
Switching from Vertical to Horizontal	209
Switching from Horizontal to Vertical	211
<b>INDEX</b>	
Label Index	215
General Index	217





# SCOPE

## **THE NEXT GENERATION OF COMPUTING**

Congratulations! With the Corvus Concept Personal Workstation, you have entered the next generation of computing.

We call the Corvus Concept a Personal Workstation because it was first conceived as a user interface to our local network, the Corvus Omninet. Adding the Concept to a network with other Concepts or different brands of computers takes only a simple connection of twisted pair cable. It is as easy as hooking up a hi-fi speaker.

The Concept can also serve as a powerful stand-alone computer; its advanced 16-bit microprocessor, swivel-screen display capabilities, and other built-in features make it an ideal operator's interface for word and data processing.

## **PURPOSE AND ORGANIZATION**

This guide is intended to give a thorough overview of the Concept. It is intended as an introduction to this system, and as a reference guide to be used whenever a question arises pertaining to the Concept.

The guide is divided into eight chapters covering the following subjects:

**CHAPTER 1**—Instructions for power-on, log-on, and setting the system time and date.

**CHAPTER 2**—Familiarization with the Concept keyboard and display monitor.

**CHAPTER 3**—In-depth instruction on managing files.

**CHAPTER 4**—Explanation of window management, and instruction on creating new windows and manipulating them.

**CHAPTER 5**—Details the methods of loading, creating, and editing character sets.

**CHAPTER 6**—Shows how to spool, manage volumes, and use the VOLUTIL utility.

**CHAPTER 7**—Covers advanced features, such as user-defined function keys, data communications, executable files, etc.

**CHAPTER 8**—Explains making peripheral connections, boot types, and changing screen orientation.

To assist in making this a useful reference guide there are two indexes. The first is an index of all the function key labels. The second is a general index of topics, and commands.

## CONVENTIONS

The word “Type” is used throughout this guide to mean that two or more characters are to be entered on the Concept keyboard. All words, symbols, spaces and punctuation beginning with the first character to the right of the word type should be typed exactly as shown. Do not add or delete punctuation at the end of a statement, and type all spaces that appear within the statement. For example,

**Type** /CCUTIL/FUTIL

would request “/CCUTIL/FUTIL” to be typed; the spaces between the word type and the first character to its right (a slash here) should not be typed. Characters may be entered in either upper or lower case.

The word “Press” is used throughout this guide to mean that a single character or key-top symbol is to be entered on the Concept keyboard. For example,

**Press** RETURN

requests the carriage return be pressed on the keyboard. When a key-top symbol is used, press the key to which it refers; do not type out the individual letters of the word shown within the key-top symbol.

Function key labels are used like keytop symbols. There are ten function keys at the top of the Concept keyboard, numbered F1 through F10. The functions of these keys display across the bottom of the Concept screen, and are also graphically depicted throughout this guide.

The function key labels change with each program. Within a program, each function key may represent up to four separate functions. The first function is obtained by pressing the function key. For example, from the Dispatcher labels, “Press **SetVol**” equates to pressing the **F4** key.

The second function is obtained by holding down the **SHIFT** key and simultaneously pressing the appropriate function key.

The third function is obtained by holding down the **COMMAND** key and simultaneously pressing the appropriate function key. For example, from the Dispatcher labels, “Press **MountMgr**” equates to simultaneously pressing the **COMMAND** key and the **F9** key.

The fourth function is obtained by holding down the **COMMAND** key and the **SHIFT** key while simultaneously pressing the appropriate function key. For example, from the Dispatcher labels, “Press **Reboot**” equates to simultaneously pressing the **COMMAND** key, the **SHIFT** key, and the **F10** key.

Function key instructions in this guide are always given in terms of the labels at the bottom of the Concept screen, and not by the key number. An instruction such as “Press **ListVol**” leaves no doubt that the function to be performed is listing a volume; if the instruction said “Press **F5**,” which is the function key corresponding to the **ListVol** label, this guide would give no clear indication of what function is being performed.

Another term used frequently in this guide is default. There are many programs and functions in the Corvus Concept Operating System (CCOS) that require responses to be entered from the keyboard. In many instances the program displays the most common (or recommended) answer to the prompt. The answer that the program offers is called the default, and it can be entered by pressing **RETURN**. If the default is not to be used, simply type the new answer and it will be used by the program.



# STARTING OUT

# 1

This chapter is devoted to powering on the Concept, log-on procedures, and setting the system time and date.

If the Concept has not already been installed, refer to "The Corvus Concept Personal Workstation Installation Guide," and follow its instructions for setting up the Workstation. If the system also needs to have a drive setup and initialized proceed to "The Corvus Concept Disk Drive Installation Guide." After the Workstation is setup, return to this section and continue with this guide.

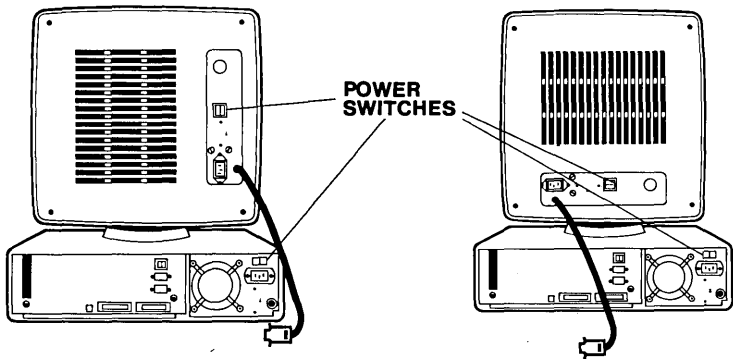
## **POWER-ON**

At this time, assuming that all the Concept hardware has been properly setup, the system is ready to power-on.

**WARNING! DO NOT POWER-ON OR POWER-OFF THE CONCEPT BASE UNIT WHEN A FLOPPY DISKETTE IS IN THE DISKETTE DRIVE. DOING THIS MAY DESTROY INFORMATION STORED ON THE DISKETTE.**

Power-on the Concept display unit. If the display unit is in the horizontal position, depress the left side of the power switch. If the display unit is in the vertical position, depress the top of the power switch.





To power-on the Concept base unit, depress the right side of the power switch on the back of the base unit. The Concept makes a beep, and enters its self-diagnostic program.

## SELF-DIAGNOSTIC

The Concept runs a 30 second self-diagnostic program after power-on. The screen is blank during the test, and the Concept beeps again when the program is finished. Either the log-on prompt or a prompt asking for the type of boot displays at this time. The difference is determined by the settings of the boot switches inside the base unit.

This time a prompt asking for the type of boot displays on the screen:

```
CORVUS CONCEPT INITIALIZATION (#.#)
COPYRIGHT 1982 CORVUS SYSTEMS, INC.
ALL SYSTEM TESTS PASSED

SELECT ROOT DEVICE (D,F,L,O):
```

If the Concept is connected to an Omninet network, press O in response to the prompt. If the Concept is connected to a local hard disk drive, press L. To boot from a floppy diskette press F, and if the Concept has the MACSbug debugger PROMs and you want to enter the debugger press D.

If there is some problem with the setup, or with the Concept itself, a boot error may appear. The following is a list of boot errors and what should be done if one appears.

<b>Boot Error #</b>	<b>System Failure</b>	<b>Solution</b>
1	UART failure	
2	Prom checksum	Call an authorized Corvus service center
3	Static RAM	
4	Dynamic RAM	
5	Dynamic RAM	
6	Slot device not ready	
7	Duplicate Omninet host numbers	Set the Omninet dip switches to a unique address

In the future, the boot device prompt can be skipped by setting the boot switches for the device from which the Concept is to boot. To do this, refer to the section titled "Type of Boot" in Chapter Eight.

## LOG-ON

The screen clears after the type of boot has been selected. Then the log-on prompt appears:

```
*
  *
                                     CORVUS SYSTEMS
                                     *
                                     CONSTELLATION II
                                     *
                                     V #.#
                                     *

PLEASE ENTER YOUR NAME:
```

Type the name that has been assigned to you, and press **RETURN**. Another prompt quickly displays:

```
PLEASE ENTER YOUR PASSWORD:
```

Type your password and press **RETURN**. If you do not have a password, just press **RETURN**. Your password does not appear on the screen when you type it.

The screen clears, then displays several rows of dots as the system information is being loaded into memory. The cursor pauses several times while they are being printed. The screen clears again and the following displays:

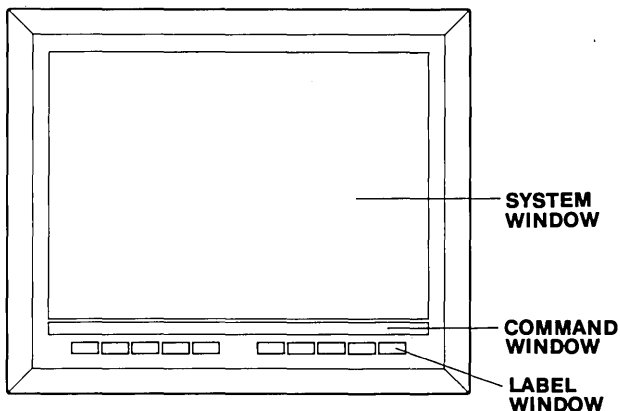
```
CCOS v #.# User: JOE Station: 1 Volume: /CCSYS
Mount Constellation II volumes ...
Mounting volume ***** on unit 9
Mounting volume CCUTIL on unit 10
Mounting volume CCSYS on unit 4
Mounting volume ONE on unit 11
Mounting volume TWO on unit 12
-----
Enter Command:
... ..
```

At the top of the screen is a permanent display which has the version of the operating system on the left, followed by the name of the user, the Workstation number (which appears only if the Workstation is connected to an Omninet network), and the current volume. The date and time (in military time) display on the right side of the screen.

All the volumes in your access table are automatically mounted. If there is a diskette drive connected to the Concept without a diskette in the unit, a series of asterisks are used in place of the diskette name when "Mounting volume \*\*\*\*\* on unit 9" appears.

If there is a diskette drive with a diskette in the unit, the diskette's name appears where the asterisks are.

The screen has been divided into three separate areas called "windows." Output displays in the System window. Interaction with the operating system takes place in the Command window. Function key labels are shown in the Label window.



The Label window shows the name of the functions that the ten function keys represent. The first labels to appear are called the Dispatcher Labels. Other sets of labels can be reached by pressing certain function keys.

The Dispatcher label display shows only one half its labels at a time. By holding down the **COMMAND** key on the keyboard the other half can be seen. With both shifted and unshifted labels, plus those made available by holding down the **COMMAND** key, a maximum of forty labels are available per level. The following shows the Dispatcher labels:

### DISPATCHER

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
p-System	LogiCalc	EdWord	SetVol	ListVol	ClrWindow	SelWindow	Restart	Help	Userkeys

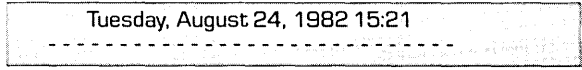
### DISPATCHER with **COMMAND** held down

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
						TmpWindow			Reboot
FileMgr	WindowMgr	ExecFile	Const II	SysUtils	RevBkgnd	CreWindow		MountMgr	

\* LogiCalc is optional. Userkeys does not appear on F10 unless conditions are met.

## SETTING THE SYSTEM DATE

SetDate sets or displays the current date. When the Concept is on, a date displays above the upper right corner of the System window.



When the Concept is first installed the date must be set to the current date and the time must be set to the current time. Once the Concept time is set, it updates the date every twenty-four hours.

To set the date enter the System Utilities level. This is done by holding down the **COMMAND** key and pressing **SysUtils** (the F5 key).

**Press** **SysUtils**

The function key labels change, and look as follows:

**SYSTEM UTILITIES**

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
									Exit
AssgnDrv		SetDtaCm	SetDate	SetTime	EdChrSet			Help	

**SYSTEM UTILITIES with **COMMAND** held down**

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
DnrVrsn									

Pressing **SetDate** displays the current date in the Command window.

**Press** **SetDate**

The Command window displays the following:

```
Today is 24-Aug-82
Select SysUtils function:
```

To set the date, enter the date as shown below then press **SetDate**. The date must be entered in the following format:

DD-MMM-YY

DD is the day, MMM is a three letter abbreviation for the month, and YY is the year. Valid alphabetic abbreviations for months are as follows:

January	—Jan	July	—Jul
February	—Feb	August	—Aug
March	—Mar	September	—Sep
April	—Apr	October	—Oct
May	—May	November	—Nov
June	—Jun	December	—Dec

In the following example the date is set to June 6, 1983.

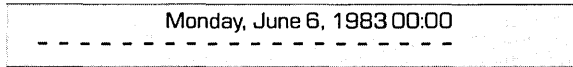
**Type** 6-JUN-83

**Press** **SetDate**

SetDate confirms the change.

```
Date set to 6-Jun-83
Select SysUtils function:
```

The new date also displays at the top right corner of the screen.



Every element of the date does not have to be typed every time the date is set. If the month and year are already correct, type the day and press . If the year is already correct, type the day, hyphen, the month and press .

For example, if the date is set to 12-Jul-83 and today is July 13, set the new date as follows:

**Type** 13  
**Press**

The date changes immediately. Now set the actual date so that the system will be accurate.

## SETTING THE SYSTEM TIME

SetTime sets or displays the current time. When the Concept is first installed the current time needs to be set. Once the time is set it should not have to be set again, because the clock continues to run even if the Concept is powered off. The date should never have to be set again either, since the Concept clock updates the date every twenty-four hours; it even takes leap years into consideration.



To show the current time,

**Press** `SetTime`

The Command window displays the following:

```
The time is 14:59:04
Select SysUtils function:
```

To set the time, type the time then press `SetTime`. The time must be entered in the following format:

HH:MM

The HH represents hours in military time (0 - 23). The MM is minutes.

In the following example the time is set to 9:31 a.m.

**Type** 9:31

**Press** `SetTime`

SetTime confirms the change in the Command window.

```
Time set to 09:31:00
Select SysUtils function:
```

In the following example the time is set to 4:59 p.m.

**Type** 16:59

**Press** `SetTime`

Again, SetTime confirms the change.

```
Time set to 16:59:00
Select SysUtils function:
```

Now enter the current time so that the Concept will be accurate.

To return to the Dispatcher level press the **Exit** function key.

You should now see the correct time displayed at the top right corner of the screen.



# SYSTEM OVERVIEW

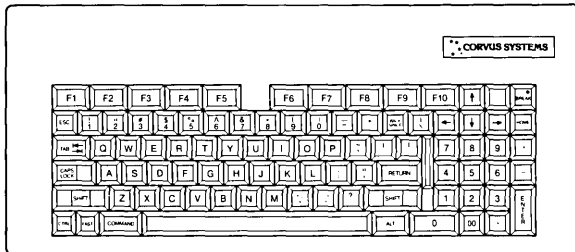
# 2

This chapter covers the Concept keyboard and display monitor. The use of the function keys, modifier keys, cursor keys, and display control keys are also discussed.

## THE KEYBOARD

The standard Concept keyboard is a detached Selectric style QWERTY keyboard. It is connected to the base unit with a telephone-type cord, which can extend to six feet in length.

Although the keyboard is similar to a typewriter keyboard, the Concept has a numeric keypad and additional keys to control some of the functions of the Workstation.



The keys are divided into five groups:

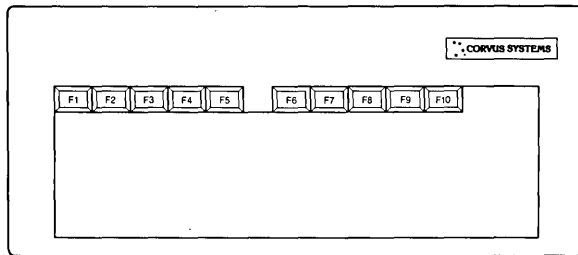
1. Character keys,
2. Function keys,
3. Cursor control keys,
4. Numeric keypad,
5. Modifier keys.

All keys on the keyboard automatically repeat when they are held down.

The keyboard has fourteen modifier keys, and ten function keys. These keys appear within boxes, e.g. **RETURN** and **COMMAND**, when they are referred to throughout this guide.

## Function Keys

There are ten function keys across the top of the Concept keyboard labeled F1 through F10. These keys are used by system programs to perform specific functions. The keys can also be user-defined for more versatility.



The function key meanings may change with each program. Within each program, each function key may represent up to four separate functions at one time. The first function is obtained by pressing the function key. For example, from the Dispatcher labels, "Press **SetVol**" equates to pressing the **F4** key.

The second function is obtained by holding down the **SHIFT** key and simultaneously pressing the appropriate function key. There is no second level function in the Dispatcher level.

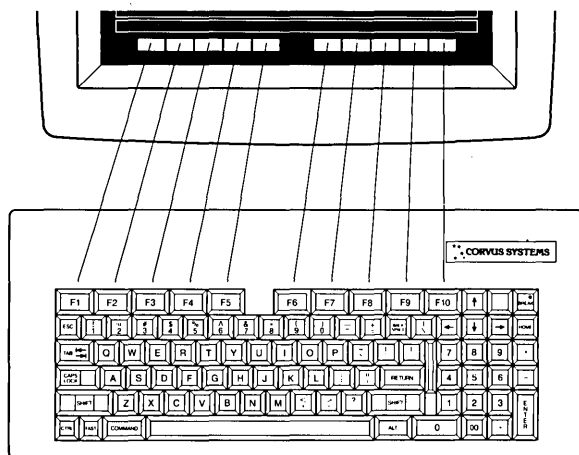
The third function is obtained by holding down the **COMMAND** key and simultaneously pressing the appropriate function key. For example, from the Dispatcher labels, “Press **MountMgr**” equates to simultaneously pressing the **COMMAND** key and the **F9** key.

The fourth function is obtained by holding down the **COMMAND** key and the **SHIFT** key while simultaneously pressing the appropriate function key. For example, from the Dispatcher labels, “Press **Reboot**” equates to simultaneously pressing the **COMMAND** key, the **SHIFT** key, and the **F10** key.

### **Function Key Labels**

Function key labels are the abbreviated names of the functions they perform. The functions can be accessed by pressing the function key corresponding to the label.

Function key labels display in the Label window at the bottom of the display screen. Each of these keys corresponds to one of the function keys, F1 through F10, on the Concept keyboard. The number of the corresponding function key displays above each label.



Two sets of function key labels are available. One set of labels displays automatically; the other set displays when the **[COMMAND]** key is held down.

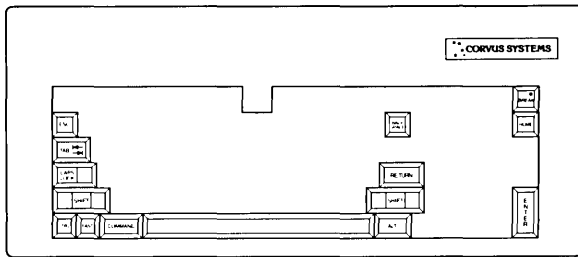
Some sets of function key labels contain a row of shifted and unshifted functions. The unshifted label is the lower one in each label box, and the shifted label is the upper one. However, not every label box has both a shifted and unshifted label; most have only the unshifted label.

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
SHIFTED LABELS					SHIFTED LABELS				
unshifted labels					unshifted labels				

This guide treats a label as a key. When a label name is enclosed in a box, e.g. **[ListVol]**, this guide refers to the function key associated with the label. When a label name is not enclosed, e.g. ListVol, the task associated with the label is being referred to, not the physical key.

## Modifier Keys

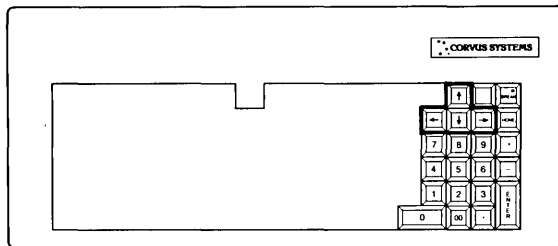
Most of the modifier keys change the function of other keys on the keyboard. When one is held down and another key is pressed, that key reacts differently than it would without the modification. Most of the modifier keys can be found on other computer keyboards, and some, like **SHIFT**, are found on typewriters too.



## Cursor Keys

The Concept keyboard has four cursor movement keys which move the screen cursor up, down, left, and right. The arrow on the keytop indicates the direction the cursor moves.

The cursor keys are located just above the numeric keypad.





## THE VIDEO DISPLAY UNIT

The Corvus Concept display unit can be arranged in different ways to suit different purposes. The orientation can be changed, and it can be tilted and swiveled. For details on changing the screen orientation from vertical to horizontal, or vice versa, see the section titled "Changing the Screen Orientation" in Chapter Eight.

### Interrupting Screen Display

If the Concept is set for scrolling, there are times when you will want to stop the flow of output. To do this hold down the `CTRL` key and press S.

A `CTRL`-S can be used to interrupt scrolling a text file, or long volume listing; it can also suspend the output of a program.

To restart a listing or program's output use the `CTRL`-S or `CTRL`-Q commands. Either key may be used to continue scrolling.

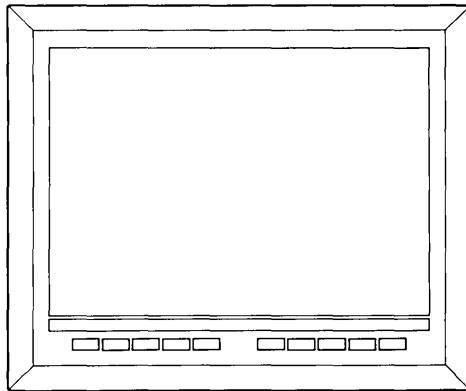
Using `CTRL`-S does not terminate a task. Before using the Concept to perform another task, the first task must be completed by restarting the display, and completing the program's execution.

### Screen Configuration

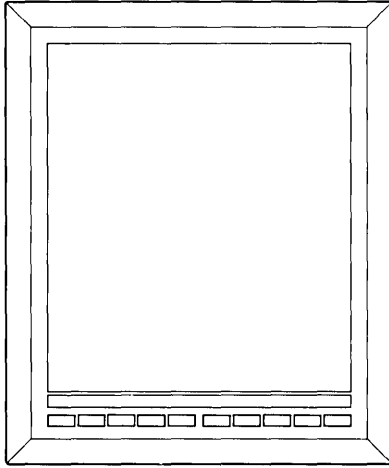
The video display screen is divided into areas called windows. A window can be considered a section of the screen, or a defined display area.

At log-on the Concept displays three permanent windows. They are called the System window, the Command window, and the Label window.

**System Window.** The System window is the topmost and largest of the three windows. This is where all normal output displays, e.g. characters and graphics.



In the horizontal orientation, the size of the System window is 707 pixels by 479 pixels.



In the vertical orientation, the size of the System window is 545 pixels by 639 pixels.

The following table is a list of the actual number of characters, of different heights and widths, that can display within the default System window.

### Text Columns On Screen

<b>Char Width in Pixels</b>	<b>Horizon.</b>	<b>Vertical</b>
5	141	109
6	118	91
7	101	78
8	88	68
9	79	61
10	71	55
11	64	50
12	59	45
13	54	42
14	51	39
15	47	36
16	44	34

### Text Lines On Screen

<b>Char Hgt. in Pixels</b>	<b>Horizon.</b>	<b>Vertical</b>
5	96	128
6	80	107
7	68	91
8	60	80
9	53	71
10	48	64
11	44	58
12	40	53
13	37	49
14	34	46
15	32	43
16	30	40

The character set being displayed, the default character set, is 6 by 10 pixels. Therefore in the horizontal orientation 118 characters can be displayed across on 48 separate lines. In the vertical orientation 91 characters can be displayed across on 64 separate lines.

**Command Window.** The Command window is the area where commands are entered and displayed. It is the long narrow rectangle below the System window. The Command window also serves as a message window.



The Command window is composed of two lines. The bottom line is the command line, where typed commands are displayed. The top line is the message line, where the Concept displays messages.

**Label Window.** Function key labels are displayed in the Label window. This is at the bottom of the screen below the Command window. The labels in this window stand out, since the lettering is black with a white background. The labels indicate what task each function key represents.

## THE DISPATCHER

The labels that display at the bottom of the screen at boot are called the Dispatcher labels. They are the first level of functions available after log-on.

Other sets of labels can be called from the Dispatcher level that allow various features of the Concept to be used. These sets include: the FileMgr labels, the WindowMgr labels, SysUtils labels, EdWord labels, and the Userkey labels. The Userkeys label will not appear in the Dispatcher level unless certain conditions are met. Userkeys are discussed in a later chapter.

The labels available at the Dispatcher level are shown below:

DISPATCHER									
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
p-System	LogiCalc	EdWord	SetVol	ListVol	ClrWindow	SelWindow	Restart	Help	Userkeys

DISPATCHER with <b>COMMAND</b> held down									
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
					TmpWindow				Reboot
FileMgr	WindowMgr	ExecFile	Const II	SysUtils	RevBkgnd	CreWindow		MountMgr	

**ClrWindow**

clears everything from the current window.

**Const II**

enters the Constellation II level which is used exclusively by system managers. To return to the Dispatcher level from the Constellation II level press **EXIT** until the Dispatcher labels reappear.

- CreWindow** creates new windows inside the System window area. To create a new window press **CreWindow** and follow the directions given.
- EdWord** enters the Concept editor/word processor program. It is discussed in a separate manual: "The Corvus Concept EdWord User Guide."
- ExecFile** runs any Exec file entered on the command line, or runs the command interpreter by typing %. To leave the command interpreter press **CTRL**-D.
- FileMgr** enters the File Manager level. To return to the Dispatcher level from the File Manager press **Exit**.
- Help** gives brief descriptions of each function key label. Press **ESC** to exit from Help.
- ListVol** lists volumes on-line or file names in a volume. Type the volume name and press **ListVol** to list the files in a particular volume. Type / or ! for listings of volumes, devices on-line, and other volume information.
- LogiCalc** enters the optional Concept spreadsheet program. It is discussed in a separate manual: "The Corvus Concept LogiCalc User Guide."
- MountMgr** runs the Constellation II Mount Manager program which is used to manage volume access, and status. To return to the dispatcher level from the Mount Manager level press **Exit**.

`p-System`

enters the UCSD p-System runtime operating system. The current volume must be a UCSD volume. H(alt must be used to return to the CCOS from the p-System.

`Reboot`

automatically reinitializes the system without going through the diagnostic program and log-on procedures.

`Restart`

causes the CCOS to perform the last command that was executed.

`RevBkgnd`

reverses the background of the current window from black to white and vice versa.

`SelWindow`

selects a defined window and makes it the current window.

`SetVol`

sets the designated volume as the current volume. The current volume is the one where most work is done.

`SysUtils`

enters the System Utilities level. To return to the Dispatcher level from the System Utilities level press `Exit`.

`TmpWindow`

creates new windows inside the System window area. The window created is not saved, and it is not cleared at inception. To create a temporary window press `TmpWindow` and follow the directions given.

`UserKeys`

enters the user-definable function key level. These labels are defined in the file `USERKEYS.TEXT`, which is created with `EdWord`, and can be modified to contain any commands available.



**W**indowMgr

enters the Window Manager level. To return to the Dispatcher level from the Window Manager level press **E**xit.

## How to Run a Program

At the Dispatcher level, the Concept is ready to execute any commands. At this point the Command window appears as follows:

Select function:

There are two ways to run a program: 1) by typing the name of a file that contains a program, or 2) by pressing the function key corresponding to a desired program label.

To run a program designated by a function key label, press the function key associated with that label. For example, to use EdWord, the Concept Editor/Word processor, press **F3**, **E**dWord.

To run a program other than one designated by function key labels, type the name of the program and press **R**eturn.

For example, to run the Concept diskette formatting program named FUTIL type the program file name.

**Type** FUTIL  
**Press** **R**eturn

If a volume name is not specified when running a program, the CCOS first searches the current volume for the file. If the file is not in the current volume, the CCOS searches the boot volume, which is the volume named CCSYS.

To run a program that is not in the current volume, type the volume name enclosed in slashes, followed by the program name, and press **[RETURN]**. For example, to run a program named GRAPH in a volume named WORK,

**Type** /WORK/GRAPH

**Press** **[RETURN]**

### **The Restart Function**

Restart executes the last command that was executed. For tasks that are to be performed many times in succession the Restart key performs a valuable service. To execute the last command performed,

**Press** **[Restart]**

If the last task performed was executing the program /WORK/GRAPH, pressing **[Restart]** causes the program GRAPH to be executed again.

## The Help Label

For a brief description of any function key label, press **[Help]**, then press the function key that corresponds to the label in question.

For example,

**Press [Help]**

The machine reads the Help file and displays a prompt for more instructions:

```
Select the appropriate function key for HELP ..  
Press <ESC> to terminate help processing
```

**Press [ClrWindow]**

The following message displays near the top of the System window:

```
[ClrWindow] clears everything from the current  
window.
```

For information on another function, just press the appropriate key. For example, if **[ListVol]** is pressed after having pressed **[ClrWindow]** above, a ListVol help message displays below the ClrWindow help message.

**Press [ListVol]**

```
[ClrWindow] clears everything from the current  
window.
```

```
[ListVol] lists volumes on-line or file names in a volume.  
Type the volume name and press [ListVol] to list the  
files in a particular volume. Type / or ! for listings of  
volumes and devices on-line and other volume  
information.
```

Help continues to give messages until the **[ESC]** key is pressed. Pressing **[ESC]** exits the Help program.

If the **[HELP]** key is accidentally pressed, the program can be terminated before completely reading the help messages by pressing any key on the keyboard.

### **The Exit Label**

There are many sets, or levels, of function key labels in the operating system. The Exit label leaves the current set of labels and returns to one level higher until the Dispatcher level is reached.

Exit is always associated in some way with the **[F10]** key. It appears in every set of labels.

## **REBOOTING THE SYSTEM**

A boot is the process of loading the operating system, and the drivers. Rebooting clears the memory, and initializes the system. The Concept provides two methods of rebooting: the warm boot and the cold boot. Both types of boot are explained in the sections below.

At times it is useful to reinitialize the system. If neither the warm or cold boots can be used the base unit can be turned off and the system can be powered on again. This boots the system, but is not recommended when either the warm or cold boot can be used.

## Warm Boot

When at the Dispatcher level, Reboot can be used to perform a warm boot. This reloads the operating system and drivers, and skips the diagnostic program and the regular log-on procedures.

To reboot this way,

**Press** `[Reboot]`

## Cold Boot

A cold boot can be performed by holding down `[CTRL]` and pressing `[BREAK]`. This reloads the operating system and drivers, but first asks for the type of boot, user-name and password.

It is the same type of boot as is performed when first powering on the Concept. However, DO NOT reboot when running a program that writes data to the disk, because this may destroy the file being written. First exit the program, then reboot the system.

To reboot this way,

Hold down `[CTRL]` and  
**Press** `[BREAK]`

This chapter covers methods of file management. Topics include deleting files, copying files, what volumes are, and the information that can be gathered from volume listings, and file management functions.

## **VOLUMES**

A volume is a defined area of the disk drive. Like a floppy diskette, it can hold a number of files and has a maximum size. The size of a disk volume is usually much greater than that of a floppy diskette, but a floppy is a volume too.

A volume is used in order to allow certain users access to certain parts of a drive, and not others. One user could be allowed access to volumes with word processing files and be denied access to the accounting software and data base in other volumes.

To the Concept, slashes are volume name delimiters. They tell the operating system that the name following the slash is a volume name and not a file name. To specify the volume named CLYDE, you would type /CLYDE. To specify the file BUDGET.TEXT in the volume FORECAST you would type /FORECAST/BUDGET.TEXT.

## **Logical Units**

A logical unit number is simply a number in a table that corresponds to different devices attached to the Concept. There is a table in memory which contains important information about the devices on-line. The logical unit number directs programs to the corresponding hardware item in that table.

Volumes are considered physical devices, and they too must be assigned a logical unit number. In most cases the logical unit number is transparent to the user. It is rarely seen, except when dealing with volumes.

## **The System Volume**

The system volume, also called the boot volume, usually has the name /CCSYS. It contains the information needed by the Concept to boot. When the Concept is booted, the boot code always searches for the system volume on logical unit four, therefore the system volume must be mounted on unit four.

## **The Current Volume**

The current volume is the volume where work takes place. At boot, the current volume is set to the home volume. The home volume is the volume mounted on unit 5. If there is no volume mounted on unit 5, the current volume is set to the system volume.

The current volume can be set to any volume in your account. This is done using the SetVol function.

## Setting the Current Volume


SetVol appears in both the Dispatcher labels and the FileMgr labels. It designates a volume as the current volume. The current volume is the one in which most work is done. The operating system looks to the current volume by default if no volume name is specified.

To change the current volume press **SetVol**, type the name of the volume to be set, then press **RETURN**.

For example, to set the volume /VOL1 as current do the following:

**Press** **SetVol**

The Command window displays:



Set which volume to current?

**Type** /VOL1

**Press** **RETURN**

When the volume is set, the Concept responds with the following:



Current volume is now /VOL1

Once a volume is set as the current volume, it remains current until it is changed, or until the next time the Concept is rebooted.



## Listing a Volume

The ListVol function can be used to obtain various kinds of information. ListVol tells which volumes are mounted, or the file names in any volume specified. ListVol is found in both the Dispatcher and the FileMgr labels.

To list the files in a volume press **ListVol**, type a slash, the volume name, and press **RETURN**.

For example,

Press **ListVol**

The Command window displays:

```
List which volume directory?
```

Type **/VOL1**

Press **RETURN**

The following displays in the System window:

```
UCSD Volume: VOL 1, size = 1024 blocks, 19 files
CVT.TEXT      512 13-Mar-82   6:   4 text
CVT           512 10-Mar-82  10:   5 data
LINKER.TEXT   228 10-Mar-82  15:  14 text
LNKFR8NT.TEXT 380 10-Mar-82  29:   6 text
LNKINTRO.TEXT  90 10-Mar-82  35:   8 text
LIBRARY.TEXT  205 10-Mar-82  43:   6 text
OBJECTF.TEXT  247 10-Mar-82  49:  80 text
LINKMAN.TEXT  333 10-Mar-82 129:   4 text
MER6.TEXT     512 19-Feb-82 133:  24 text
MER0.TEXT     512 13-Mar-82 157:  22 text
MER1.TEXT     512 13-Mar-82 179:  34 text
MER2.TEXT     512 13-Mar-82 213:  26 text
MER3.TEXT     512 13-Mar-82 239:  24 text
MER4.TEXT     512 13-Mar-82 263:  28 text
MER5.TEXT     512 13-Mar-82 291:  24 text
BOLD          512 13-Mar-82 315: 162 data
BOLD1         512 23-Mar-82 477: 500 data
BB.TEXT       512 13-Mar-82 977:   4 text
GRAPH.TEXT    512 22-Mar-82 981:  20 text
<...>                1001:  23
19 or 19 files listed
```

Some volume listings are very long, and it may be preferable to pause the listing. Using `[CTRL]-S` suspends the listing, and `[CTRL]-S` again or a `[CTRL]-Q` resumes the listing.

To terminate a listing at any point press `[BREAK]`. This terminates the listing, and the bottom line is printed telling how many files, out of how many files in the volume, were listed.

To understand this listing it is best to break it down one piece at a time. The top line of the listing gives some important volume information:

```
UCSD Volume: VOL1, size = 1024 blocks, 19 files
```

UCSD Volume: VOL1 — UCSD says that it is a UCSD format volume. A volume could alternately be a CCOS volume. The name of the volume being listed is VOL1.

size = 1024 blocks — this says that the volume VOL1 has a size of 1024 blocks. Since each block is 512 bytes, the volume size is one half a megabyte.

19 files — this says that there are nineteen files in the volume VOL1. This includes all text and data files.

The next line from the listing, and the lines below it give valuable information about the files. The information is broken down into six columns:

```
CVT.TEXT      512 13-Mar-82   6:   4 text
```

CVT.TEXT — this column has file names in the volume directory.

512 — this column has the number of bytes used in the last block of the file.

13-Mar-82 — this was the system date the last time the file was updated. If you have set the date on your machine this is the actual date the file was last updated.

6: — this number is the starting block address of the file. The address is relative to this volume and not the entire drive.

4 — this is the number of blocks that the file occupies. If this number is added to the starting block address the result is the starting block address for the next possible file.

text — this tells whether the file is a text or a data file.

The last two lines also give useful information. They are as follows:

```
<...> 1001: 23  
19 of 19 files listed.
```

<...> — this symbol is in the filename column, and represents no file. This means that the space shown on the right is open space on the volume.

1001: 23 — this is the starting block address and the size of the open space. It tells where the open space begins, and how large the space is.

19 of 19 files listed. — this says that all the files in the volume were listed. There are occasions where a partial listing is preferable to a full one. For a partial listing of only three files, the last line says 3 of 19 files listed.

Occasionally a file appears in the listing that has the following information in the date column:

```
MESS.I 512 0-???-100 978: 344 data
```

The three question marks indicate that this is a temporary file. It is a file that is left in an open state. Most temporary files are files that are created by a program, and then are not closed for some reason. Usually they are created when a person reboots during the execution of a program, or when a program has a bug. The only thing to do with a temporary file is to delete it. See the section “Deleting a Temporary File” later in this chapter.

**Slash /** To list all the devices on-line, which includes volumes,

**Press** `ListVol`

**Press** /

**Press** `RETURN`

The following displays in the System window:

```
0: /NULL
1: /CONSOLE
2: /SYSTEM
4: # /CCSYS      2047 blks
5: # /VOL1       1024 blks
6: /PRINTER
9: # /FLOPPY     500 blks
11: # /CSD       1024 blks
12: # /CCUTIL    2048 blks
14: # /VOL2      1024 blks
15: # /VOL3      1024 blks
18: # /MYUCSD    1024 blks
22: # /GRAPH     1024 blks
30: /SLOT10
31: /DTACOM1
32: /DTACOM2
33: /OMNINET
34: /TIMER
35: /KYBD
36: /DISPLAY

System volume is: /CCSYS
Current volume is: /VOL1
```

To understand the slash type of listing a closer analysis of one line is needed. For this description the following line from the previous listing is used:

```
14: # /VOL2 1024 blks
```

- 14: — this is the logical unit number.
- # — this indicates the device that follows is block structured (i.e., a volume).
- /VOL2 — this is the name of the device. In this case it is a volume named VOL2. Others are /DTACOM1 for the serial data port 1, and /PRINTER for the printer, etc.
- 1024 blks — this is the length of the volume in blocks. It is the same number that would be seen on the top line of a volume listing. It is on only a few of the lines in this listing, because not all of the devices are volumes.

The bottom lines add some information regarding the present status of the system.

```
System volume is: /CCSYS  
Current volume is: /VOL1
```

The first line says that the system volume is /CCSYS which we can see in the listing is on logical unit four, and is 2048 blocks long.

The second line says that the current volume is /VOL1 located on logical unit number five. Because it is logical unit 5 it is the current volume at boot. If another volume had been set to the current volume before doing the ListVol the new volume would be listed as current.

**Exclamation Point !** The exclamation point type of volume listing gives more volume information than the slash listing. This extra information includes the active slot number, the disk server number, the drive number, and the starting block address.

To list all devices mounted, including the volume block addresses,

**Press** `ListVol`

**Press** !

**Press** `RETURN`

The following displays in the System window:

```
0: /NUL
1: /CONSOLE
2: /SYSTEM
4: # /CCSYS      2048 blks (slt 2 srv 0 drv 1 blk 1232)
5: # /VOL1      1035 blks (slt 2 srv 0 drv 1 blk 5328)
6: /PRINTER
9: # /FLOPPY    500 blks (slt 3 srv 0 drv 0 blk 0)
11: # /CSD      1024 blks (slt 2 srv 0 drv 1 blk 6352)
12: # /CCUTIL   2048 blks (slt 2 srv 0 drv 1 blk 3280)
14: # /VOL2     1024 blks (slt 2 srv 0 drv 1 blk 7376)
15: # /VOL3     1024 blks (slt 2 srv 0 drv 1 blk 8400)
18: # /MYUCSD   1024 blks (slt 2 srv 0 drv 1 blk 9424)
22: # /GRAPH    1024 blks (slt 2 srv 0 drv 1 blk 10448)
30: /SLOT10
31: /DTACOM1
32: /DTACOM2
33: /OMNINET
34: /TIMER
35: /KYBD
36: /DISPLAY

System volume is: /CCSYS
Current volume is: /VOL1
```

The listing is the same as the slash type of listing but includes extra volume information. Below is an explanation of the added information.

```
(slt 2 srv 0 drv 1 blk 7376)
```

slt 2 — this is the slot number that the drive is connected to. If the volume is on a local hard disk drive or a diskette drive, the slot number could be from 1 to 4, depending on where the disk's interface card is connected. If the volume is on a drive connected to Omninet, the slot number would be 5.



- srv 0 — this is the disk server number. The disk server is used to connect a Corvus disk to Omninet. If the slot number is not 5, this number is irrelevant.
- drv 1 — this is the disk drive number. Because up to four disk drives may be daisy-chained the 1 represents the first drive.
- blk 7376 — this is the starting block address for the volume on the hard disk drive.

## **MANAGING FILES**

The Concept File Manager handles most of the tasks of manipulating and managing files.

These tasks include:

1. making files,
2. deleting files,
3. moving files,
4. concatenating files,
5. renaming files,
6. copying files,
7. listing files, and
8. listing volumes.

Files are usually created automatically by programs. When saving a file with EdWord, a CCOS file is automatically created. Other programs, such as LogiCalc, the compiler, and assembler, create files as well.

There are several types of files in the CCOS. The two most important types are data files and text files. Programs are placed in data files. Letters, manuals, and lists of names are in text files, which always have .TEXT automatically appended to their name by EdWord.

## The File Manager

Concept file management functions include the tasks mentioned in the section above. These tasks are performed by running the programs in the File Manager level.

When **FileMgr** is pressed from the Dispatcher level, the Concept FileMgr labels display.

**FILE MANAGER**

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
									Exit
CopyFile	DletFile	ListFile	SetVool	ListVol	ClrWindow	SetWindow	Spool	Help	

**FILE MANAGER with **COMMAND** held down**

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
						TrmpWindow			
MakeFile	DletTemp	RnamFile	ConcFile	Crunch	RevBkgnd	CreWindow	ChngVol	MountMgr	

**ChngVol**

changes a designated volume between the CCOS and UCSD volume types.

**ClrWindow**

clears everything from the current window.

**ConcFile**

combines two or more files into one continuous file. It is also used to copy a single file and assign it a different destination file name.

- CopyFile** copies a file from one volume to another. Both files are given the same name.
- CreWindow** creates new windows within the System window area. To create a new window press **CreWindow** and follow the directions given.
- Crunch** moves all files in a volume so that all unused space is combined into one large empty space at the end of the volume.
- DletFile** deletes a file, or files, from a volume. It may not be possible to retrieve a file after it has been deleted.
- DletTemp** deletes a temporary file, or files from a volume. A temporary file shows up in a volume listing with ??? where the month normally appears.
- Exit** leaves the File Manager and returns the Concept to the Dispatcher level.
- Help** gives brief descriptions of each command label. To leave Help press **ESC**.
- ListFile** lists the contents of a text file in the current window.
- ListVol** lists volumes on-line or file names in a volume. To list the files in a particular volume press **ListVol**, and type the volume name. Type / or ! for listings of devices on-line, volumes mounted and other volume information.
- MakeFile** creates a file, and asks for the name and size.

- MountMgr** runs the Constellation II Mount Manager program. Press **Exit** to return to the File Manager.
- RevBkgnd** reverses the background of the current window, from black to white and vice versa.
- RnamFile** renames a file asking for the old name and the new one.
- SetVol** sets the designated volume as the current volume. The current volume is the default volume when work is being done.
- SelWindow** selects a defined window and makes it the current window.
- Spool** runs the Constellation II spool program. This program sends files to a special volume named Pipes. These files can be despoiled by a station, with a local printer, using the **Spool** function.
- TmpWindow** creates new windows inside the System window area. The window created is not saved, and it is not cleared at inception. To create a temporary window press **TmpWindow** and follow the directions given.

One important purpose of the FileMgr functions is to help keep disk space organized and clear of unwanted files. Because only 77 files are allowed per volume, you should periodically remove files that are no longer needed. This is done using the Dletfile function.

## File Names

A file name can be from one to fifteen characters. The name may contain only letters, numbers, periods, dashes, and underline characters. It may start with any legal character. File names can be changed with the RnamFile function described later in this chapter.

The following are examples of valid file names:

REJECTS.IV.TEXT	LETTER.SORT
A__FAST__PROGRAM	56__TEXT
TRY.324.TEXT	PROGRAM669
... HERE ...	__3.FLY

The following are examples of invalid file names:

TOOLONGFILENAME	—this is too long.
BAD#SYMBOL	—the # is an illegal symbol.
!NOTTHIS?	—neither the ! or the ? are legal symbols.
NO SPACE	—no spaces are allowed in file names.

To reference a file in the current volume, simply type the file name, as follows:

SECRET.TEXT

To reference a file in a volume other than the current volume, type the volume name enclosed in slashes followed by the file name, as follows:

/HUSH/SECRET.TEXT

## **Wild Card Characters**

There are two wild card characters that can be used with some of the File Manager functions. They are the asterisk (\*) and the question mark (?).

The question mark can be used to represent any single character. For example, the file name ?OSE.TEXT is equivalent to NOSE.TEXT, HOSE.TEXT, and LOSE.TEXT. The file name FOO?.TEXT is equivalent to FOOL.TEXT, FOOD.TEXT, and FOOT.TEXT.

The asterisk represents any group of characters. For example, the file name F\*.TEXT is equivalent to FUN.TEXT, FUNNY.TEXT, FLOAT.TEXT etc.

If a volume listing is requested using /VOL1/\*.TEXT all the file names in VOL1 that end with .TEXT are listed. If a volume listing is requested using /VOL1/LET\* all file names that begin with the letters LET are listed.

The wild card characters can be used with the ConcFile, CopyFile, DletFile, DletTemp, ListFile, and ListVol commands.

## **Deleting a File**

The DletFile function found in the FileMgr labels is used to remove files from volume directories. After deleting a file, the operating system no longer recognizes it, because it has been removed from the directory.

When a file has been deleted from a directory, its directory entry can be reconstructed with the MakeFile function described later in this chapter. However, the file is restored only if other data has not been copied onto the location where the file was stored.

The following example deletes the file MESS.TEXT from the current volume, /VOL1. To remove a file from a volume that is not the current volume remember to enclose the volume name in slashes.

**Press** `DelFile`

The Command window displays:

```
Delete which file(s)?
```

**Type** MESS.TEXT

**Press** `RETURN`

The Command window displays:

```
Delete file "/VOL1/MESS.TEXT"?
```

You must press Y for yes to the prompt shown above in order to delete the file. If you do not want to delete the file press N.

**Press** Y

The Command window displays:

```
Update volume /VOL1 directory?
```

This message prompts for a yes or no answer as to whether the volume directory should be updated. You must press Y for yes if you want the file to be deleted. If you press N the file is not deleted.

### **Press Y**

The file is deleted, the FileMgr level returns, and the select function prompt reappears.

Wild card characters may be used with DletFile. If one is used in a file name with DletFile, the program checks all file names that match the wild card file name. One at a time, and in alphabetical order, the FileMgr asks if you want to remove the matching files. When you press Y in response to the prompt, the file indicated is marked to be deleted. If you press N, the file is not deleted.

**The Q Option.** To avoid being prompted for each file and the “Update volume /VOL1 directory?” prompt, the DletFile function has a quiet option that restrains these prompts. The quiet option is used by placing -Q before the name of the files that are to be deleted.

In the following example, this option is used:

**Press**

The Command window displays:



Delete which file(s)?

**Type** -Q /VOL2/MESS.TEXT

**Press**



The file is deleted, and the FileMgr level returns.

The quiet option may be used with the CopyFile, ConcFile, and DletTemp functions as well. When used with the other functions the -Q is placed at the end of the line, not the beginning. The operation is discussed more thoroughly in the sections covering those functions.

### **Deleting a Temporary File**

Temporary files are created when files are opened by a program and do not become closed. Normally, these temporary files are deleted automatically when the program has completed its task. On occasion, errors in the program being executed cause the operation to terminate before the temporary file can be automatically deleted.

It is easy to spot a temporary file in a volume listing. In the date column a temporary file has three question marks where the month should be.

```
MESS.I 512 0-???-100 978: 344 data
```

Temporary files are deleted like regular files. The DletTemp function allows asterisks and question marks as wild card characters as does DletFile.

In the example below, the temporary file MESS.I is deleted from the volume /VOL1.

**Press** `DletTemp`

The Command window displays:

```
Delete which temporary file(s)?
```

**Type** /VOL1/MESS.I

**Press** `[RETURN]`

The Command window displays:

```
Delete file "/VOL1/MESS.I"?
```

**Press** Y

The file is deleted, and the File Manager reappears.

## Recovering Lost Files

When you have accidentally deleted a file, the MakeFile function is very helpful. When a file is deleted it is not actually erased unless other information has been copied over the place where that file resided. That is, a file is not destroyed unless the Crunch command has been used, or a file has been written over the same location. When a file has been deleted from a directory, its directory entry can be reconstructed with the MakeFile function.

MakeFile creates files. When a file is made the space that it occupies is not formatted, or cleared. An entry is made in the directory saying that the information stored in that area belongs to this file. If the information is still intact, a directory entry can be created and a deleted file can be recovered.

Wild card characters cannot be used with MakeFile.

In the following example, a file twelve blocks long called BLT.TEXT is created in the current volume.

**Press** `MakeFile`

The Command window displays:

```
Make which file?
```

**Type** BLT.TEXT

**Press** `RETURN`

The Command window displays:

```
How many blocks for file "BLT.TEXT"?
```

**Type** 12

**Press** `RETURN`

When MakeFile is finished, it gives a message that the file has been made, and a ListVol shows the file entry.

When MakeFile creates a file it searches the volume directory, and then creates the file in the first empty space in the volume large enough for the size specified. If a size is not specified after the file name, MakeFile creates the file in the largest empty space in the volume and makes the file one block smaller than the empty space.

To recover a lost file, list the volume that contained the file and look for a blank space in the listing:

```
UCSD Volume: VOL1, size = 1024 blocks, 3 files
BOLD          512 29-FEB-83  6: 162 data
LETTER.TEXT   512  1-JUN-83 168: 12 test
< . . . >                                180:  42
OBJECT.TEXT   512  1-JUN-83 222: 34 test
< . . . >                                256: 768
3 of 3 files listed.
```

The deleted file appears like the line between LETTER.TEXT and OBJECT.TEXT in the listing above. The filename is replaced with < . . . >, and the only information left to identify the file is its size and address. To re-create that file enter the FileMgr and press MakeFile, like in the preceding example, and make a file of exactly 42 blocks in length (or however many blocks the deleted file happens to be). This creates a file in that space and recovers the file you might have thought was lost.

If a file was deleted at the end of a volume, that file's space would appear mixed with the empty space at the end. There would be no way to determine the size of the file if you did not have its size in blocks written down or committed to memory.

## Copying a File

The CopyFile function is used to move files from one volume to another. This function copies the source file, and creates a duplicate file in a specified destination volume.

Wild card characters and the quiet option described in the “Delete a File” section of this chapter can be used.

In the following example the file WORK.TEXT is copied from volume /VOL1 to volume /VOL2.

**Press**

The Command window displays:

Copy which file(s)?

**Type** /VOL1/WORK.TEXT

**Press**

The Command window displays:

Copy file(s) to which volume?

**Type** /VOL2

**Press**

Copy file then asks for verification:

Copy file "/VOL1/WORK.TEXT"?

**Press** Y

**Press**

The Command window displays the results of the copy, including the length of the file transferred:

```
/VOL1/WORK.TEXT copied to /VOL2/WORK.TEXT (10 blocks)
Select FileMgr function:
```

Copying multiple files is similar to copying a single file, except that the \* (asterisk) or ? (question mark) wild card characters can be used instead of, or with, a file name. Wild card characters are used in the same fashion as described previously in this chapter.

For example, all the files in /VOL1 can be copied to another volume by typing /VOL1/\* to the “Copy which file(s)?” prompt.

**Type** /VOL1/\*

The “Copy file ‘XXXXX’?” prompt asks for verification of each file one by one in alphabetical order unless the quiet option (-q) is added after the destination volume name. To copy all the files without being prompted enter the destination volume name in this way:

**Type** /VOL1 -Q

If the destination volume already has a file with the same name as the file being copied, the Concept File Manager asks which copy is to be kept. The following prompt appears:

```
Delete old "WORK.TEXT" (file dates are the same)?
```

The prompt says “. . . (file dates are different)” if the file dates are not the same. The Concept waits for a yes or no answer before continuing. In the prompt above, the old file is the one already in the destination volume, not the version of the file being copied.

**The D Option.** This option gives the files being copied the current system date. If this option is not used the old date (the last time the file was updated) remains with the file.

This option is entered with the destination volume name.

**Type** /VOL2 -D  
**Press**

**The S Option.** This option strips all escape codes from a file. If a text file was written using EdWord and included bold face or underlined characters, the S option can strip the enhancements from the text. This is useful when trying to print a file to a printer that does not support these features.

The S option is entered with the destination volume name.

**Type** /VOL2 -S  
**Press**

The Q, D, and S options are all placed on the line with the destination volume. Their order of placement is not important when more than one is used. An example is shown below:

**Type** /VOL2 -S -Q -D  
**Press**

## Concatenating Files

ConcFile performs two functions: it concatenates several files into one new file, or it copies a single file to a file with a new name. In contrast to CopyFile, the name of the destination file must be specified.

Concatenating files means that the files are combined into one file. The second file named is appended to the first file. Wild card characters may be used when entering the source file names.

The following example shows how to combine two files, PORK.TEXT and BEANS.TEXT, in the volumes /VOL1 and /VOL2 respectively. BEANS.TEXT is appended to PORK.TEXT and the output is to a file called PNB.TEXT in /VOL3.

**Press**

The Command window displays:

Concatenate which files?

**Type** /VOL1/PORK.TEXT /VOL2/BEANS.TEXT

**Press**

The Command window displays:

Output concatenated files to where?

**Type** /VOL3/PNB.TEXT

**Press**



The next example shows how to use the asterisk wild card to concatenate files. In this example, all the files in the volume /VOL2 that match the file name LIST\*, according to wild card rules, are concatenated. Output is to a file called /VOL2/LST.TEXT

**Press** `[ConcFile]`

The Command window displays:

Concatenate which files?

**Type** /VOL2/LIST\*

**Press** `[RETURN]`

The Command window displays:

Output concatenated files to where?

**Type** /VOL2/LST.TEXT

**Press** `[RETURN]`

In the following example, the file NOTES.TEXT in the current volume is copied to the file NTS.TEXT in the volume /VOL3.

**Press** `[ConcFile]`

The Command window displays:

Concatenate which files?

**Type** NOTES.TEXT

**Press** `[RETURN]`

The Command window displays:

Output concatenated files to where?

**Type** /VOL3/NTS.TEXT

**Press** [RETURN]

## Renaming a File

RnamFile renames files. Wild card characters cannot be used with RnamFile.

To rename a file, press [RnamFile], type the file name then press [RETURN]. Next type the new name for the file, and press [RETURN].

In the following example, the file OLD.TEXT in the current volume, /VOL1 is changed to NEW.TEXT.

**Press** [RnamFile]

The Command window displays:

Rename which file?

**Type** OLD.TEXT

**Press** [RETURN]

The Command window displays:

What is new name for file "OLD.TEXT"?

**Type** NEW.TEXT

**Press** [RETURN]

The Command window displays:

File "/VOL1/OLD.TEXT" renamed to "/VOL1/NEW.TEXT"  
Select FileMgr function:


## Listing a File

ListFile displays the contents of a text file in the current window. Wild card characters can be used with ListFile.

In the following example, the file NOTES.TEXT in the current volume is listed.

**Press** `[ListFile]`

The Command window displays:



List which files?

**Type** NOTES.TEXT

**Press** `[RETURN]`

The System window displays the contents of the file.

Once the listing starts, it can be suspended with `[CTRL]-S`; either a `[CTRL]-S` or a `[CTRL]-Q` can be used to restart the listing. To terminate a file listing, press `[BREAK]`.

When wild card characters are used with ListFile all files that are applicable are listed in alphabetical order.

## Redirecting Output

Normal output displays on the video display unit. This output can be redirected to a text file or to a printer. Redirection works with any of the Concept listing functions: ListVol, ListFile, and DrvrVrsn (discussed later).

The printer driver must be assigned before redirecting output to a printer. See the next section for loading the printer driver.

The most common place to redirect output is to a printer. To redirect a volume listing to a local printer,

**Press** `[ListVol]`  
**Type** `/VOL1 > /PRINTER`  
**Press** `[RETURN]`

The format for redirecting output is as follows:

`[function key]`(device being listed) > (output device)

The device being listed can be a file or a volume. The output device can be a text file or a printer.

The following example shows how to redirect the volume listing to a text file in another volume.

**Press** `[ListVol]`  
**Type** `/VOL1 > /VOL2/LIST.TEXT`  
**Press** `[RETURN]`

## Printing a File

There are two ways to print a file: with a local printer attached, or by spooling the file to a shared printer on a network. Spooling is covered in a later chapter, and is not discussed in this section.

Before a file can be printed the printer must have a driver assigned to it. This can be done in just a few steps. From the Dispatcher level,

**Press**

**Press**

**Press**

If the printer attached to the Concept is an NEC 7710 printer press  until the Dispatcher level returns. The printer driver defaults to the settings necessary for the NEC. If the printer is either an IDS 460 or an Epson MX-100,

**Press**

The function key labels appear as follows:

PRINTER TYPE									
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Epson	Nec	IDS							NoChange

Press the function key corresponding to the type of printer you have. Then press  until the Dispatcher level reappears.

If you are using a different type of printer than one of the ones shown above, you may have to consult the section titled "Setting Data Communication Parameters."

The printer driver should be set now, and be capable of performing the functions in the previous section on redirecting output. A file can be printed using the ListVol function and redirecting output to the printer. The format for that command would be as follows:

**Press** `ListFile`

**Type** `/VOL2/DOCUMENT.TEXT > /PRINTER`

**Press** `RETURN`

Printing of text files on a local printer is usually done from within the text editor/word processor program called EdWord.

## **Editing a File**

To create or edit a document there is EdWord, the editor/word processor program, which can be called from the Dispatcher level. Before entering the program there are a few steps that should be considered.

### **Loading the 7 by 11 Character Set.**

EdWord was written to take advantage of the display capabilities of the high resolution screen. It displays bold face, and other enhancements, but only if a character set 7 by 11 dots or larger is loaded. The default character set is 6 by 10 dots, and is too small to display bold and other enhancements.


To load the 7 by 11 dot character set return to the Dispatcher level. From the Dispatcher level,

**Press** **[WnDowMgr]**

The function key labels change and one appears with the label **[LdDispCh]**. This stands for Load Display Character Set.

**Press** **[LdDispCh]**

The Command window displays the following:



Enter name of character set file:

**Type** /CCUTIL/CSD.07.11

**Press** **[RETURN]**

The character set is loaded. Press **[Exit]** to return to the Dispatcher level.

### **Creating An EdWord Workspace.**

EdWord has its own file structure within the CCOS. The workspace is similar to a volume, but the CCOS recognizes it as a file within a volume.

Before actually entering EdWord and creating the workspace the volume must be checked to make sure that there is enough space for a workspace. A workspace is usually quite a large file. The default size of an EdWord workspace is 494 blocks, which is the smallest size recommended. This size is equal to approximately 84 pages of text.

To be sure there is enough space, list the volume where the workspace is to be created. Look at the free space at the end of the listing, and be sure that it is larger than the size of the workspace to be created. If there are any gaps of free space in the listing, the space can be recovered using the Crunch function discussed in the section titled "Reclaiming Volume Space." Then set the volume as current.

To enter EdWord and create a workspace,

**Press** `[EdWord]`

The Command window displays the following:

```
Workspace name? W
```

**Press** `[RETURN]`

EdWord looks for a workspace already named W and the following prompt appears in the Command window:

```
There is no Workspace "W." Make a new one? N
```

The default is no, so to make a new workspace,

**Press** Y

The next prompt quickly appears:

```
Name of new workspace? W
```



Whatever name was entered to the first prompt, "Workspace name? W" also appears here. If the default was accepted then the name W is the default here also. To create a workspace of another name type the name and press **[RETURN]**. To accept the default,

**Press** **[RETURN]**

The next prompt is the final step in creating a workspace.

A screenshot of a terminal window showing a prompt: "Creating workspace. Number of blocks? 494". The text is displayed in a monospaced font on a light background.

The default size is enough for approximately 84 pages of text. This should be suitable for most purposes. To accept the default,

**Press** **[RETURN]**

The workspace is created and an introductory tutorial to EdWord is placed in the current workpad. After looking at this file and playing with EdWord you will be familiar with some of the commands available. To learn more, consult "The Corvus Concept EdWord User Guide."

**Loading a File.** The procedure for loading a text file into the workspace is simple. The labels in EdWord appear as they do below:

### EDWORD

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Enhance	Format	Redo	Bound	Ins Line	Del Line	Adjust	Userkeys	JoinLine	Exit
ScrlBack	ScrlFwr	Undo	Mark	Ins Char	Del Char	Erase	Search	Workpad	Go To

### EDWORD with **COMMAND** held down

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Forms									Verify
CutText	PasteLin	PasteOvr	PasteCol	CopyText	Print	Flush L	Center	Flush R	Set

To load a file,

**Press** **Workpad**

The labels change again:

### Workpad Level

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
		Del Pad			Make Dir		List Vol		
LoadFile		ClearPad		SaveFile	Make Pad	Ins File	View Dir	View Pad	Exit

**Press** **LoadFile**

The Command window displays the following prompt:

Read which file?

Now type the name of the file to load into EdWord, and press **RETURN**. EdWord accepts only TEXT files and appends .TEXT to all files, so do not type the .TEXT that is part of the file name.

For example, if the file name is LETTER.TEXT simply type LETTER. If LETTER.TEXT is entered EdWord looks for LETTER.TEXT.TEXT which is too long a file name. If this happens EdWord asks "Read which file?" again. If any problems arise consult "The Corvus Concept EdWord User Guide."

**Saving a File.** If you are creating a program you must create the source file in EdWord, and then save it out to the CCOS. The TEXT file that is created in the CCOS is the file that is then compiled.

To save a file from EdWord enter the Workpad level. To do this,

**Press**

From the Workpad level,

**Press**

The Command window displays the following prompt:

At this point a default file name appears that is the same name that the file was created with, or the last file name that the file was either read from or written to. Either press  or type the name that you wish the file to be given. EdWord creates a new file for you if there is not already a file by that name. EdWord appends .TEXT to whatever name is entered, so be sure the file name that is entered is ten characters or less.

If the name you enter is more than ten characters long, EdWord asks “Write to which file?” again. If any problems arise consult “The Corvus Concept EdWord User Guide.”

**Exiting EdWord.** Exiting EdWord is a simple process. Be sure to return to the main level by pressing the Exit key from any level. Then from the main level press **Exit** . It is the only Exit key in the shifted position in EdWord.

**Press** **Exit**

Re-entering the Workspace. To re-enter the workspace,

**Press** **EdWord**

The Command window displays the following prompt:



Workspace name? W

If the workspace you created does not have the default name of W, type the name of the workspace. If it is not in the current volume, be sure to include the volume name.

**Press** **RETURN**

After specifying the name of a valid workspace, you are immediately placed in the same workpad, or document, and at the same location that you were in when you last exited that workspace.

If you have any difficulties consult “The Corvus Concept EdWord User Guide.”

## Changing a Volume

The ChngVol command, located in the File Manager, switches the volume type from UCSD to CCOS and vice versa. The key flips the volume type back and forth if the same volume is designated each time.

To change the volume /VOL3 from CCOS to UCSD,

**Press** `[ChngVol]`

A prompt appears in the command window:

```
Change which volume?
```

If /VOL3 is the volume to be changed,

**Type** /VOL3

**Press** `[RETURN]`

The volume is switched from CCOS to UCSD the program leaves the following message as it returns to the File Manager level:

```
Volume /VOL3 is now UCSD format.  
Select FileMgr function:
```

## Reclaiming Volume Space

Crunch, found in the File Manager, reclaims unused volume space. It moves all the unused space to the end of the volume, compacting all files to the beginning of the volume.

The following example shows how to crunch a volume named /VOL2.

**Press** `[Crunch]`

The Command window displays:

```
Crunch which volume?
```

**Type** `/VOL2`

**Press** `[RETURN]`

To crunch the current volume simply press `[RETURN]` to the prompt in the line above. The screen displays the file names in the System window as they are being moved and when the crunch is done the program says that the volume has been crunched:

```
Moving CRUMB1.TEXT  
Moving CRUMB2.TEXT  
Moving MUNCH.TEXT  
Moving BITS.TEXT  
Moving CRUMB3.TEXT  
Volume /VOL2 crunched
```



# MANAGING WINDOWS | 4

This chapter focuses on the functions found in the Window Manager level. Creating windows and manipulating their environment is discussed.

## **WINDOW MANAGEMENT**

Windows are operator or program defined areas of the Concept display screen. The current window displays all normal output. Three windows are defined by the operating system at boot: the System window, the Command window, and the Label window. As far as the user is concerned these windows are permanent.

The Concept provides the means to create and manage several windows of various sizes and dimensions. Up to seventeen windows can be created, each with different dimensions, and if desired each can have a different character set. Each window is a rectangle, and must be within the confines of the original System window.



## The Window Manager

The following are the Window Manager functions: create window **[CreWindow]**, delete window **[DelWindow]**, select window **[SelWindow]**, clear window **[ClrWindow]**, reverse window background **[RevBkgnd]**, change scroll **[ScriMode]**, load keyboard character set **[LdKybdCh]**, load display character set **[LdDispCh]**, create temporary window **[TmpWindow]**, and run test patterns **[TestPtrn]**.

Pressing **[WindowMgr]** from the Dispatcher enters the Window Manager level. The function keys change meaning again, and the labels appear as follows:

WINDOW MANAGER									
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
LdDispCh		LdKybdCh			ClrWindow	SelWindow	ScriMode	Help	Exit

WINDOW MANAGER with <b>[COMMAND]</b> held down									
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
					RevBkgnd	CreWindow	DelWindow	TestPtrn	
						TmpWindow			

- [ClrWindow]** clears everything from the current window.
- [CreWindow]** creates new windows within the System window area. To create a new window press **[CreWindow]** and follow the directions given.
- [DelWindow]** deletes the current window. After selecting the window to be deleted, press **[DelWindow]**.
- [Exit]** returns the system to the Dispatcher level.

- Help** gives brief messages for each function key pressed. Help is left by pressing the **ESC** key.
- LdDispCh** loads a display character set. This is discussed in the chapter titled “Character Sets.”
- LdKybdCh** loads a keyboard character set. This is discussed in the chapter titled “Character Sets.”
- RevBkgnd** reverses the background of the current window from black to white, and vice versa.
- ScrlMode** changes the screen display from scroll mode to page mode and vice versa. In page mode **CTRL-S** must be pressed for each new page.
- SelWindow** selects a defined window and makes it the current window.
- TestPtrn** enters a test pattern program to display test patterns on screen.
- TmpWindow** create a temporary window inside the System window. The window is not cleared at inception. To create a temporary window press **TmpWindow** and follow the directions given.

## Clear Window

ClrWindow clears the current window. This function also appears in the Dispatcher labels and the FileMgr labels.

When ClrWindow is used, everything within the bounds of the window is erased; i.e. graphics and text.

To clear the current window,

**Press** `ClrWindow`

When a window is clear, display output starts at the top of the window; otherwise, it starts where the last task left off.

## Reverse Background

RevBkgnD changes the screen background of the current window from black to white, or from white to black. This function also appears in the Dispatcher labels and the FileMgr labels.

To reverse the background of the current window,

**Press** `RevBkgnD`

If the background is white, press `RevBkgnD` to change it to black. If the background is black, press `RevBkgnD` to change it to white.

## Scroll Mode

With ScrlMode, the Concept can be set to display listings one of two ways: scrolling, or paging.

One method of displaying is the scroll mode. At boot the Concept is set for scroll mode, which means scrolling occurs one line at a time. If the text being displayed is longer than the window length, scrolling pushes all lines of text up. The top line is pushed off the top of the window, and a new line displays at the bottom of the window. This process continues until all the text has been scrolled.

The other method of display is the page mode. This means that text displays one window at a time. If the text being displayed is longer than the window length, displaying stops when it reaches the bottom of the window. To continue this type of display, press **[CTRL]-Q** or **[CTRL]-S**.

If the Concept is set for scroll mode, pressing **[ScrlMode]** changes it to page mode. The following message displays in the Command window when **[ScrlMode]** is pressed:

```
Window is now in page mode
Select WindowMgr function:
```

If the Concept is set for page mode, pressing **[ScrlMode]** changes it back to scroll mode. A message displays in the Command window:

```
Window is now in scroll mode
Select WindowMgr function:
```

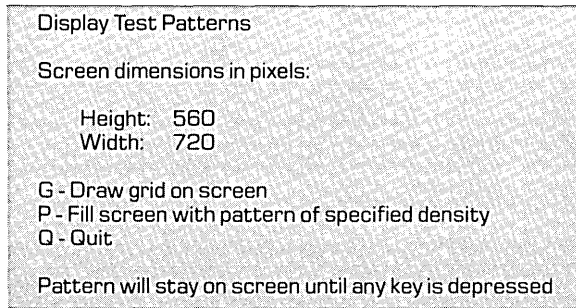
## Displaying Test Patterns

The TestPtrn label in the Window Manager level tests the Concept display screen by drawing grids or shading the screen with different pixel densities.

From the Window Manager labels,

**Press** `TestPtrn`

The entire screen clears, then the following displays:



To draw a grid,

**Press** G

The program prompts for a number:

Number of grids [2 - 40]:

Type any number between two and forty. A grid with that number of rows and columns displays on the screen.

To fill the screen with a specific pixel density,

**Press** P

The program prompts for a number:

```
Pixel frequency:
```

Type any number in response. The screen fills with pixels according to the density specified. For example, if 1 was entered every pixel would be illuminated. If 5 were entered, every fifth pixel would be illuminated, etc.

## Creating a Window

In addition to the three permanent windows up to seventeen user-defined windows can be created with CreWindow. This function also appears in the Dispatcher labels and the FileMgr labels.

To create a window, press `CreWindow`, move the cursor to where you want the upper left corner of the new window and press `RETURN`, then move the cursor to where you want the lower right hand corner of the new window and press `RETURN`.

The CreWindow function measures windows in character units the size of the character set in the current window. The following example shows how to create a window 34 characters long and 22 characters high.

**Press** `CreWindow`

The following message displays in the Command window:

```
Move cursor to upper left corner and press <return>
```

Use the cursor keys to move the cursor. Move the cursor to a point near the upper left corner of the screen.

**Press** `RETURN`

The Command window displays:

```
Move cursor to lower right corner and press <return>
```

Move the cursor down two spaces. The Command window displays:

```
New window size is 1 by 3  
Move cursor to lower right corner and press <return>
```

Continue moving the cursor down until the Command window displays:

```
New window size is 1 by 22  
Move cursor to lower right corner and press <return>
```

Now move the cursor to the right until the Command window displays:

```
New window size is 34 by 22  
Move cursor to lower right corner and press <return>
```

**Press** `RETURN`

When a window is created, the operating system automatically selects that window as the current window, draws a box around it, and clears it. As the current window, it is where all program output displays. A window remains current until another window is selected, created, or the system is rebooted.

If the second corner of the new window is along the same line as the first corner, an error message displays on the top line of the Command window.

```
Invalid Bounds  
Select WindowMgr function:
```

If the above message displays, try again with different dimensions.

## Selecting a Window

SelWindow selects the window in which program output is to be displayed. When a window is selected, it becomes the current window and is outlined with a box. This function also appears in the Dispatcher labels and the FileMgr labels.

The System window can be selected or any other window that has been created. At least one other window must have been created for this function to have any use. SelWindow selects the windows in the order in which they were created.

To select the next window,

**Press** `SelWindow`

Once the last window has been selected, the next time `SelWindow` is pressed the permanent System window is selected again.



## Deleting a Window

DelWndow deletes the current window. However, this window must be one that was created, not the permanent System window.

When a window is deleted it no longer exists and cannot be selected until it is re-created.

To delete a window,

**Press** `SelWndow`

until the window to be deleted is selected as current. Then,

**Press** `DelWndow`

Neither the System window, the Command window, nor the Label window can be deleted.

## Creating a Temporary Window

In addition to the three permanent windows and the seventeen user-defined windows, TmpWndow creates one temporary window. It can be found in Dispatcher and File Manager levels as well.

The procedure for creating a temporary window is exactly the same as the procedure described in the previous section for creating windows.

The difference between a temporary window and a permanent window is that when a temporary window is created, the operating system automatically selects that window as the current window, draws a box around it, but DOES NOT clear it. Also, as soon as another window is selected as the current window, the temporary window is deleted.

## **PRINT WINDOW**

A function found in the volume /CCUTIL is the Print Window utility. It is listed in the directory as PRTWND, and prints whatever appears in the current window.

Before the Print Window utility may be used a local dot matrix printer must be attached to the Concept, and the printer driver has to be assigned.

The utility is used by creating a temporary window that encompasses the area of the screen that is to be printed. The create temporary window command, `[TmpWindow]`, is found in the Dispatcher level, the Window Manager level, and the File Manager level. Its use is explained in the section immediately before this.

PRTWND can be used without a temporary window, and can print the entire System window if you choose. The reason for making a temporary window is so that the program will run faster. It is also useful for printing part of the screen.

The second step, after having set the window, is to type the command.

**Type** /CCUTIL/PRTWND

**Press**

The window being displayed is printed on a local dot matrix printer. This utility cannot spool to a shared printer or a file.

This chapter discusses the methods of loading, creating, and editing display character sets. The character set editor is discussed in depth.

## CHARACTER SETS

The Concept is capable of displaying a large variety of character sets. The character sets that are provided with the system are in the volume /CCUTIL and begin with the prefix CSD.

There is one other display character set; it is in /CCSYS, and is called CSD.DEFAULT. This file contains the default character set that is used in all three windows at boot. It is the same as the set called CSD.06.10 in the volume /CCUTIL.

Character sets are developed within a certain range of pixels. Pixels are the dots that make up the screen. Each character is made up of a collection of these dots within a certain area. Each character in the set CSD.06.10 is described within a range of six pixels across by ten pixels high. The size of the set must be determined before the characters can be created. The maximum range is from one to sixteen in each direction.

Each window of the Concept can be set to display a unique character set. The character set displaying in a window can be changed from the default by using a command in the Window Manager, LdDispCh. This command loads a different display character set into the window.

A character set can contain up to 192 elements, or characters. Most, including the default set, have only 96 characters. If a character set has 192 characters, the second 96 characters are displayed when the **[ALT]** key is held down and a key is pressed. Character sets that have 192 characters are called alternate character sets. The alternate character sets normally have the letters .ALT appended to their name.

## **LOADING A CHARACTER SET**

Loading a character set is simple. From the Dispatcher labels, press **[WindowMgr]**. The two function keys labeled LdDispCh and LdKybdCh are used to load character sets. They are each discussed below.

### **Loading a Display Character Set**

LdDispCh loads a display character set in the current System window.

To do this select the window in which the character set is to display. In the following example the character set CSD.09.14.ALT is used. It is found in the volume /CCUTIL.

**Press** **[LdDispCh]**

The Command window displays:

```
Enter name of character set file:
```

**Type** /CCUTIL/CSD.09.14.ALT

**Press** `[RETURN]`

LdDispCh displays a message when the set is loaded:

```
Display character set /CCUTIL/CSD.09.14.ALT loaded
Select WndowMgr function:
```

## Loading a Keyboard Character Set

LdKybdCh loads a keyboard character set other than the default. The Concept does not come with an alternate set. The only set is the QWERTY set CSK.DEFAULT corresponding to the keyboard provided.

In the following example the character set in the file CSK.DEFAULT is loaded.

**Press** `[LdKybdCh]`

The Command window displays:

```
Enter name of keyboard character set file:
```

**Type** CSK.DEFAULT

**Press** `[RETURN]`

LdKybdCh displays a message when the set is loaded:

```
Keyboard character set: CSK.DEFAULT loaded
Select WndowMgr function:
```

## ALTERNATE CHARACTER SETS

The Corvus-supplied alternate character sets contain characters for most of the 192 possible keys. The second 96 characters contain modified vowels and symbols. The modified vowels are arranged so that, as far as possible, their positions can be remembered easily. The correspondence between shifted and unshifted characters is the same.

To use the alternate characters either CSD.09.14.ALT, CSD.07.11.ALT, or CSD.06.10.ALT must be loaded as was shown in the previous section. In this example the set CSD.09.14.ALT, the largest of the three alternate character sets, is used.

Once the set is loaded the alternate characters can be used by holding down the **[ALT]** key and pressing any key.

In general the rules for using alternate characters are that holding down **[ALT]**:

- and pressing a vowel produces the unlauded version of that vowel.
- and pressing the preceding alphabetical character to the vowel produces the grave accented version of that vowel. An exception to this rule is O since N is used for Ñ. To get the grave accented O, use M.
- and pressing the following alphabetical character to the vowel produces the acute accented version of that vowel.
- and pressing the second succeeding alphabetical character to the vowel produces the circumflex version of that vowel.

## The 9 by 14 Alt Character Set Table

The tables on the following two pages show the decimal number, hexadecimal number, keyboard symbol and character produced. The second table is the upper half of the alternate character set. To use these keys the **ALT** key must be held down. These charts can be very helpful when using the character set editor, as well as on other occasions when the alternate character set is used.



**9 by 14 alt Character Set Table**

DEC	HEX	KEYTOP	CHAR	DEC	HEX	KEYTOP	CHAR
32	20	SPACE	SPACE	80	50	P	P
33	21	!	!	81	51	Q	Q
34	22	"	"	88	52	R	R
35	23	#	#	83	53	S	S
36	24	\$	\$	84	54	T	T
37	25	%	%	85	55	U	U
38	26	&	&	86	56	V	V
39	27	"	"	87	57	W	W
40	28	(	(	88	58	X	X
41	29	)	)	89	59	Y	Y
42	2A	*	*	90	5A	Z	Z
43	2B	+	+	91	5B	[	[
44	2C	,	,	92	5C	\	\
45	2D	-	-	93	5D	]	]
46	2E	.	.	94	5E	'	'
47	2F	/	/	95	5F	_	_
48	30	0	0	96	60	,	,
49	31	1	1	97	61	a	a
50	32	2	2	98	62	b	b
51	33	3	3	99	63	c	c
52	34	4	4	100	64	d	d
53	35	5	5	101	65	e	e
54	36	6	6	102	66	f	f
55	37	7	7	103	67	g	g
56	38	8	8	104	68	h	h
57	39	9	9	105	69	i	i
58	3A	:	:	106	6A	j	j
59	3B	;	;	107	6B	k	k
60	3C	<	<	108	6C	l	l
61	3D	=	=	109	6D	m	m
62	3E	>	>	110	6E	n	n
63	3F	?	?	111	6F	o	o
64	40	@	@	112	70	p	p
65	41	A	A	113	71	q	q
66	42	B	B	114	72	r	r
67	43	C	C	115	73	s	s
68	44	D	D	116	74	t	t
69	45	E	E	117	75	u	u
70	46	F	F	118	76	v	v
71	47	G	G	119	77	w	w
72	48	H	H	120	78	x	x
73	49	I	I	121	79	y	y
74	4A	J	J	122	7A	z	z
75	4B	K	K	123	7B	{	{
76	4C	L	L	124	7C	:	:
77	4D	M	M	125	7D	}	}
78	4E	N	N	126	7E	'	'
79	4F	O	O	127	7F	-	-

9 by 14 alt Character Set Table

DEC	HEX	KEYTOP	CHAR	DEC	HEX	KEYTOP	CHAR
160	A0	SPACE	SPACE	208	D0	P	Ó
161	A1	!	í	209	D1	Q	Ô
162	A2	"	í²	210	D2	R	Ø
163	A3	#	£	211	D3	S	Å
164	A4	\$	°	212	D4	T	Ü
165	A5	%	°	213	D5	U	Û
166	A6	&	ª	214	D6	V	Ü
167	A7	'	L	215	D7	W	Û
168	A8	(		216	D8	X	Œ
169	A9	)		217	D9	Y	Ɔ
170	AA	*	†	218	DA	Z	ç
171	AB	+	‡	219	DB		š
172	AC	,		220	DC		š
173	AD	-	—	221	DD		μ
174	AE	.	·	222	DE		π
175	AF	/	↑	223	DF	—	°
176	B0	0	—	224	E0	'	à
177	B1	1	ß	225	E1	a	á
178	B2	2	½	226	E2	b	â
179	B3	3	⅓	227	E3	c	ã
180	B4	4	¼	228	E4	d	ä
181	B5	5	ω	229	E5	e	è
182	B6	6	α	230	E6	f	é
183	B7	7		231	E7	g	ê
184	B8	8		232	E8	h	ë
185	B9	9		233	E9	i	ì
186	BA	:	Σ	234	EA	j	í
187	BB	;	†	235	EB	k	î
188	BC	<		236	EC	l	æ
189	BD	=	=	237	ED	m	ò
190	BE	>	.	238	EE	n	õ
191	BF	?	¿	239	EF	o	ö
192	C0	@	À	240	F0	p	ó
193	C1	A	Á	241	F1	q	ô
194	C2	B	Â	242	B2	r	ø
195	C3	C	Ã	243	F3	s	å
196	C4	D	Ä	244	F4	t	ù
197	C5	E	Å	245	F5	u	ü
198	C6	F	Æ	246	F6	v	ú
199	C7	G	È	247	F7	w	û
200	C8	H	É	248	F8	x	œ
201	C9	I	Ê	249	F9	y	ÿ
202	CA	J	Ë	250	FA	z	ß
203	CB	K	Ì	251	FB	{	«
204	CC	L	Í	252	FC		«
205	CD	M	Æ	253	FD	}	»
206	CE	N	Ï	254	FE		σ
207	CF	O	Ö	255	FF	Sh. BS	

## EDITING A DISPLAY CHARACTER SET

Corvus provides a character set editor so that anyone can create their own character set or modify an existing one. The EdChrSet label is found in the SysUtils level.

To use the character set editor follow along with the example below. The character set CSD.09.14.ALT is used.


### Entering the Character Set Editor

To run the character set editor,

**Press** `[SysUtils]`


**Press** `[EdChrSet]`

A window is created immediately. It is the Parameter window, and will display the information needed to start using the editor. At this time all it shows is the version number of the character set editor:



Character Set Editor, V.122

The Command window prompts for the file name of the character set. An existing character set file is being used for this example. However, if no name is entered at this point the editor asks in the Command window "New File? Y." Pressing `[RETURN]` again creates a new file, and starts by prompting for the cell size.



Character set file:

**Type** /CCUTIL/CSD.09.14.ALT

**Press** `[RETURN]`

After typing the name of the character set file, EdChrSet displays the current character set parameters for this file in the Parameter window.

```
Width: 9
Height: 14
First char: 32
Last char: 255
```

In the Command window, EdChrSet asks if the parameters are to be changed.

```
Character set file: /CCUTIL/CSD.09.14.ALT
Change parameters? Y
```

A response of N for No, causes EdChrSet to use the previous parameters for the set. A response of Y for Yes, causes EdChrSet to prompt for the new parameters. To learn how to change the parameters accept the default,

**Press** `RETURN`

The Command window is cleared, and the new parameter prompts display one at a time in this window. The first prompt is for the dot height of the character cell.

Each individual character is contained in what is called a cell. It is a rectangle of dots, or pixels, that make up the shape of the character. The cell can be as large as 16 x 16 dots.

The previous dot width for CSD.09.14.ALT was 9. We will change this parameter to 11.

## Type 11

**Press**

```
Width (1 - 16)? 11
```

The next prompt is for the dot height of the character cell. The previous dot height for CSD.09.14.ALT was 14. We will not change this.

**Press**

```
Height (1 - 16)? 14  
Width (1 - 16)? 11
```

EdChrSet prompts for the first character number. This is the decimal number which represents the character. The first character is usually 32, which is the previous first character number for CSD.09.14.ALT. We will not change this parameter either.

**Press**

```
Height (1 - 16)? 14 First (32 - 255)? 32  
Width (1 - 16)? 11
```

The next prompt is for the last character number. This number can be up to 255, which is what CSD.09.14.ALT was set at. This may be changed to any number between 32 and 255 as long as the number is greater than the first character number just entered.

We will cut this file in half by changing this to 127 for the example. Do not worry! This change affects only the copy of this file being used, not the copy the Concept has saved on the disk drive. When we are done we can save this file under a new file name, and CSD.09.14.ALT will be the same as before.

**Type** 127

**Press**

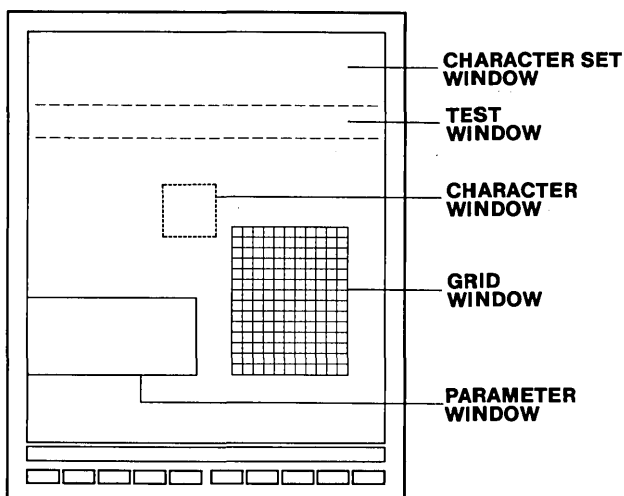
```
Height (1 - 16)? 14 First (32 - 255)? 32
Width (1 - 16)? 11 Last (32 - 255)? 127
```

When this is done EdChrSet asks if these are the parameters that you really want to work with. Pressing N, causes EdChrSet to start again from the "Width?" prompt. Since Y is the default prompt,

**Press**

```
Height (1 - 16)? 14 First (32 - 255)? 32
Width (1 - 16)? 11 Last (32 - 255)? 127 OK? Y
```

As soon as  is pressed, EdChrSet creates five separate windows inside the System window. The new windows are the Grid window, the Character window the Character Set window, the Test window, and the Parameter window.



The new parameters are displayed in the Parameter window, the Grid window displays a grid of the dimensions specified, and the Character Set window displays the character set CSD.09.14.ALT.

The labels appear as follows:

CHARACTER EDITOR									
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Test		SaveFile		Select					Exit

**Exit**

gives an opportunity to save the file before returning to the SysUtils level.

**SaveFile**

asks for the name of the file that the character set is to be saved to. It provides the original filename as the default.

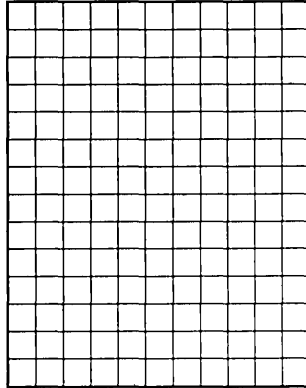
**Select**

places the character that the cursor is on at that moment into the character grid, and enters the edit mode.

Test

enters the test mode. From there any new characters can be tried in combination, and the character enhancements can be used.

An 11 by 14 grid is displayed in the Grid window.



The character set is displayed at the top of the screen in the character set window. The cursor displays at the upper left corner of this window. The cursor arrow keys are used to move the cursor to the character to be changed.

The Command window displays the ASCII number of each character. The number changes as the cursor moves. Move the cursor to the capital J character. The following displays in the Command window:

ASCII 74

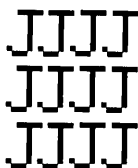


## The Edit Mode

To work on a character, move the cursor to the capital letter J, and

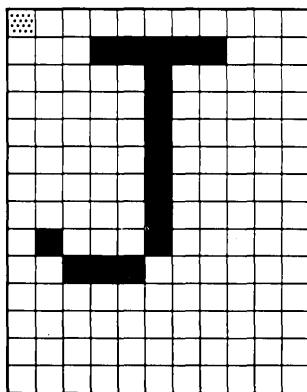
Press

Three things happen: 1) twelve real-sized copies of the letter display in the Character window; 2) an enlarged version of the letter displays in the Grid window; and 3) a new set of labels displays at the bottom of the screen. The Character window is shown below:



The Character window shows what the character looks like at normal size.

The Grid window displays the dot arrangement of the character.



The new set of labels are the editing labels. A brief description of them is given here:

### EDITING LEVEL

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
				Ins Line	Del Line		Set Col	Set Line	
Save Ch	Abandon		Copy Ch	Ins Col	Del Col	Clear Ch	Clr Col	Clr Line	Flip

**Abandon**

leaves the select mode, without saving the character.

**Clear Ch**

clears the entire Grid window so that a character can be created from scratch.

**Clr Col**

turns off all the pixels in the column, changing them to black.

**Clr Line**

turns off all the pixels in the line, changing them to black.

**Copy Ch**

asks for a keyboard character. That character is entered into the Grid window.

**Del Col**

deletes the column that the cursor is presently on, and moves all columns to the right of the cursor one space to the left.

**Del Line**

deletes the line that the cursor is presently on, and moves all lines below the cursor up one row.

**Flip**

changes the color of the pixel the cursor is on. It turns the pixel on and off like a light switch.

**Ins Col**

inserts a column where the cursor is, and all columns from the cursor position to the right are moved one column to the right.

**Ins Line**

inserts a line where the cursor is, and all columns below the cursor are moved down one row.

**Save Ch**

saves the pattern that is in the Grid window.

**Set Col**

turns all the pixels in the column on, turning them white.

**Set Line**

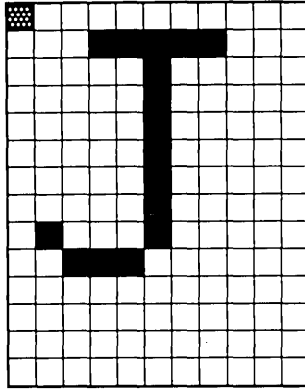
turns all the pixels in the line on, turning them white.

A special cursor is displayed in the upper left corner of the Grid window. To move this cursor use the cursor arrow keys on the keyboard.

Before moving the cursor,

**Press** **Flip**

The square containing the cursor turns white. (The images in the diagrams have the opposite color of the actual screen. This is because of the black background on the screen, and the white color of the paper.)



Notice that this change occurs in the Character window also. All changes made in the Grid window are reflected in the Character window.

JJJJ  
JJJJ  
JJJJ

To change the corner box back to black, press **Flip** again.

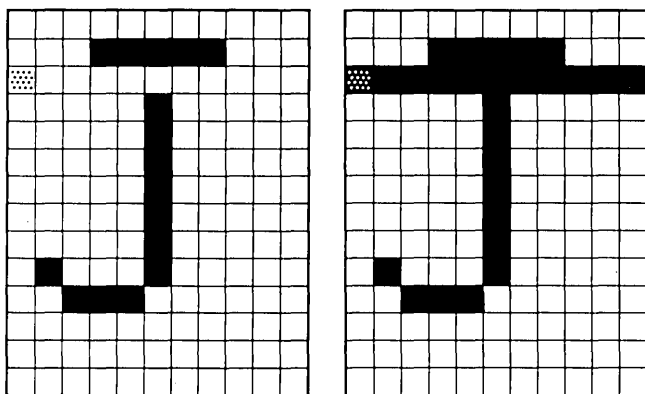
Now move the cursor down two rows, so that it is just below the bar on the top of the J. It is very easy to manipulate the shape of a character as you will see if you follow along with these instructions.

Press **Ins Line**

The entire body of the character should move down one row. The figure should appear like the one on the left below. To fill in the line,

**Press** `Set Line`

The line should fill in completely, and the grid should look like the one below and on the right.



Now if you preferred it without the line filled in,

**Press** `Clr Line`

The figure should return to its previous form. If that was not enough of a corrective measure the Del Line function will reverse the action previous to that.

**Press** `Del Line`

The line is deleted and the J should appear as it did at first.

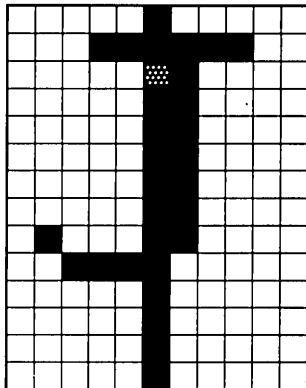
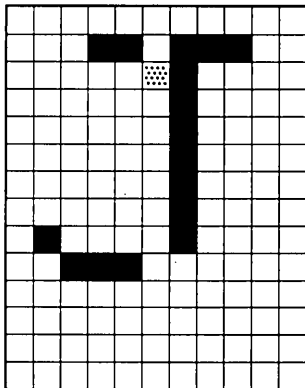
Now try moving the cursor to the right five spaces. The cursor should be on the vertical line of the J.

**Press** `Ins Col`

The J is split by this action. Everything from the cursor right is moved one column to the right as it appears in the picture below.

**Press** `Set Col`

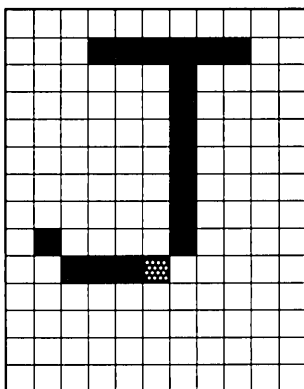
The column that the cursor is on should now be lit. The character in the grid window is looking less and less like the letter J now, and should appear like the character in the picture on the right.



There is a Clr Col key for the Set Col function just as there was the Clr Line for the Set Line.

**Press** `Clr Col`

The J is taking its previous steps in reverse. Rather than deleting the column, make the J wider. Move the cursor up one line and press **Flip**. Then move the cursor down eight lines, so that it is even with the bottom of the J, and press **Flip** again. The J should appear as it does below.



Now move on to a different character. The wide J is not really worth keeping at this time, so the edit mode can be left without saving the character.

**Press** **Abandon**

The old J and the cursor are now in the Character Set window, and the labels have returned to their original level.

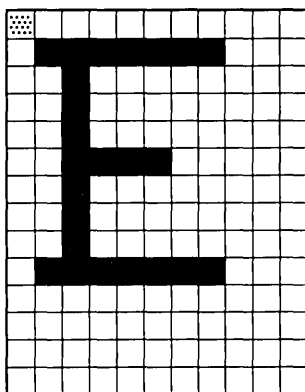
Move the cursor to the letter E.

**Press** **Select**

The capital letter E is now in the Grid window, and the labels have returned to the edit mode. The Command window tells the cursor position:

Editing ASCII 69

The grid window appears as below:



If you want to rearrange the characters in the file, you could begin by moving the J into the E character space.

**Press** `Copy Ch`

**Press** J

The Command window displays the following message briefly as the character is being changed.

Copying ASCII 74



The character that is being edited is still the E, but at this time it appears as the letter J. Since there is already a letter J, clear the grid and start with a clean slate.

**Press** `ClearCh`

The entire grid clears. Try moving the cursor from place to place, and use the Flip, Set Line, and Set Col commands. See what sort of character you can create.

If you like the character save it.

**Press** `SaveCh`

The new character and the cursor are now in the Character Set window where the letter E used to be. You are now in the select mode again.

## **Saving a Character Set File**

If you want to save the file at this point,

**Press** `SaveFile`

A prompt is displayed in the Command window asking for the file name you wish to save the character set in.

```
Filename: /CCUTIL/CSD.09.14.ALT
```

The default file is the file name of the file being edited. Usually this is the file to save to, but in this case you do not want to, since the letter E has been changed to some other figure and the parameters have been changed.

**Type** /VOLI/CSD.MYSET

**Press** RETURN

The file is created and the character set has been saved.

## Using the Test Mode

Another useful key at this level is the Test key. This key enters the test mode, so that characters can be seen in combination with others.

**Press** Test

The cursor moves to the test window, and a new set of labels appears.

<b>TEST</b>									
<b>F1</b>	<b>F2</b>	<b>F3</b>	<b>F4</b>	<b>F5</b>	<b>F6</b>	<b>F7</b>	<b>F8</b>	<b>F9</b>	<b>F10</b>
Bold	UndrLine	Dbi UndL			ClrWindow	RevBkGnd			Exit

Bold

makes all the characters typed appear in bold face. The second time it is pressed turns off the bold face.

ClrWindow

clear the test window of all characters.

Dbi UndL

places a double-thick line under each character that follows. The second time it is pressed it is turned off.

Exit

leaves the test mode and returns to the select mode.

**RevBkGnd**

reverses the background from white to black and vice versa.

**UndrLine**

places a line under each character typed.  
The second time it is pressed it is turned off.

To see how large the Test window really is,

**Press** **RevBkGnd**

Now the Test window and the Character Set window are clearly defined. Type any characters that you want. If you type a capital E, the character you created appears. Try some of the character enhancements next.

**Press** **Bold**

**Type** This is in BOLD FACE.

**Press** **RETURN**

**Press** **Bold**

The Test window should look as follows:

This is in **BOLD FACE**.

**Press** **UndrLine**

**Type** This is underlined.

**Press** **RETURN**

**Press** **UndrLine**

The Test window should look as follows:

This is in **BOLD FACE**.  
This is underlined.

**Press** **Dbl UndL**

**Type** This is double-underlined.

**Press** **RETURN**

**Press** **Dbl UndL**

The Test window should look as follows:

```
This is in BOLD FACE.
This is underlined.
This is double-underlined.
```

To leave the test mode,

**Press**

This causes the Concept to return to the main menu.

### **Exiting the Character Set Editor**

To leave the character set editor,

**Press**

This returns you to the System Utilities level, but first the editor displays a prompt regarding the file:

```
Save character set file on disk? Y
```

If you type N for No, the editor prompts once more to ask if you are sure. To save the file, press .

**Press** N

**Press** Y

**Press**

Otherwise the editor prompts you as it does below:

```
Filename? /VOL1/CSD.MYSET
```

Here you can type a new name for the character set, so that the old one is not affected.

**Type** /VOL1/CSD.THESET

**Press**

The System Utilities labels reappear, and the Dispatcher level can be reached by pressing  once.

This chapter discusses four programs that interact with Corvus Winchester disk drives. They are the Mount Manager, Spool and Despool programs, plus VOLUTIL a system utility program.

## **SPOOLING AND DESPOOLING**

One of the main purposes of a local area network is to allow a number of persons to share disk files and peripheral equipment. The Corvus Spool and Despool programs can be used to perform these tasks.

Once a file has been sent, or spooled, to the Pipes volume it remains there until it is removed, or despoiled. When a file is despoiled from the Pipes area the destination may be a printer, a file, or the console.

The Despool program allows for a Concept, or other brand of computer that is Corvus disk compatible, to remove files that have been sent to the Pipes area. The computer can be dedicated to despooling information as it arrives, acting as a printer server.

## Spooling a File

Spool is in the File Manager level. To run the Spool program,

**Press** `[FileMgr]`

**Press** `[Spool]`

The following displays in the System window:

```
Searching for network PIPES area . . . .
```

The Spool program looks for the network Pipes area, and if it is found it enters the program. If not, it looks for any local Pipes area. When the Pipes area is found the following menu appears on the screen:

```
Spooling to slot 5, server SERVER

Valid options are:

Start      — Start spooling
TextFile   — Text file . . . . . {$I}
NewPage    — New page . . . . . PG
PipeName   — Spool pipe name . . . . PRINTER

Message    — File message . . . . .Route to: WILLIE
                                     October 14, 1983 11:15 am
Enhance    — Strip enhancements .YES
Exit       — Exit program

AltSlot    — Set alternate slot
Status     — Current pipe status
```

To leave a function at any time while running the Spool program press the `[ESC]` key.

**Spool Options.** Some of the default options may have to be changed before spooling a file. The spooling options tell the program where page breaks are, when another file is to be included, if it should ignore all formatting, what the pipe name should be, whether or not to strip enhancements from the text, and active slot for spooling.

The following are descriptions of each of the options:

- Start - Begins the Spool process.
- TextFile - Yes means that the file being spooled needs carriage returns and linefeeds. No indicates that the file is not to be formatted, and should be spooled as is.
- Include - This can be any string of three characters. It is used to include other files in the file being spooled. At the point of the include string, the spooler includes the file specified, and then returns to the point in the original file where it left off.
- NewPage - This can be any string of three characters. If matching characters are found in the text file being spooled, a form feed character is placed in the spooled file.



PipeName - This can be any name up to eight characters long. The default name of PRINTER is normally used for sending files to the Pipes area that are to be printed.

If the file being spooled is not a text file, TextFile should be changed to No. To do this

**Press**  TextFile

Since there are no choices to be made, TextFile automatically changes to NO, and the formatting options Include and NewPage disappear. If TextFile is pressed again it changes back to YES and the options reappear.

When spooling a text file the TextFile option should be YES.

The Newpage string is the default. When spooling a file created by EdWord the NewPage option does not have to be changed. The setting for the NewPage string must be the same as the string used in the file being spooled. If however, you want to change the Newpage string,

**Press**  NewPage

The following appears in the Command window:



```
Set to blank to turn off new page check
Enter new page prefix:
```

**Type** {\$I

The NewPage string is changed in the menu.

The Include option allows files to be chained together or inserted within other files. The include option can be any string of three characters. The setting for the include string must be the same as the string used in the file being spooled.

The include string may be placed on any line in the file. The name of the file to be included must be specified on the same line as the string. The include string can also be placed at the end of the file. After the file has been included the spooler returns to the original file and continues to spool the remaining lines.

To change the include string,

**Press** `[Include]`

The following appears in the Command window:

```
Set to blank to turn off include check
Enter include prefix: {$}
```

**Type** `.IN`

**Press** `[RETURN]`

The include string is changed in the menu. The following is an example of how an include string can be used in a text file:

```
. . . At this point I would like to show you a copy of the
memo that was sent to all employees on December 7.
.IN /VOL1/MEMO.TEXT
As you can see, the memo . . .
```

The file MEMO.TEXT in volume /VOL1 would be included in the spool after the line containing "...December 7." The spool would then continue with the line "As you can see..."

To allow the spooler to send enhanced characters to the printer, or another file, the Enhance option must be set to NO. The default, YES, means that it will strip any enhancements from the text. The text is not changed, except for the loss of bold, and underlined features. To change Enhance to NO,

**Press**  Enhance

The field changes automatically back and forth from YES to NO and vice versa, each time  Enhance is pressed.

To change the file message that prints on the header sheet of the file,

**Press**  Message

The Command window displays the following:

```
Enter spooled file message: Route to: WILLIE Oct...
```

The default message includes the user name, and the date and time from the system. This message can be changed by typing anything and pressing  RETURN.

The last spool option is the PipeName. The default pipe name is PRINTER. It can be any string up to eight characters in length. A different pipe name may be used when there are two or more shared printers, or if you intend to despool to a file.

To change the pipe name,

**Press**

The following appears in the Command window:



Enter new pipe name: PRINTER

**Type** LETTER

**Press**

The pipe name is now LETTER, and it appears in the menu.

**Spool.** Now that the options are set the file can be spooled.

**Press**

The following prompt appears in the Command window:



Enter text file name [ TEXT ]:

**Type** /VOL1/NEW\_\_FILE

**Press**

The file is spooled to the pipename LETTER. While it is being spooled a series of dots appear across the screen and are followed by a message when the spool is complete:

```
Spooling file to pipe LETTER[ 1]
.....
.....
.....
15 blocks written to pipe LETTER[ 1]
```

The file has now been spooled, and is waiting to be despoiled from the Pipes volume.

### **Despooling a File**

Now that the file has been spooled to the Pipes volume it needs to be retrieved with the Despool program. Setting up a Concept to despool to a printer is described in "The Corvus Concept System Manager's Reference Guide."

Despooling to a file is a different procedure, and is explained here.

Before a file can be despoiled to a file, a file must be spooled to a different pipe name than PRINTER. In the previous section the file /VOL1/NEW\_FILE was spooled to the pipe name LETTER. The file is not automatically despoiled to the printer, and is waiting to be manually despoiled.

To enter the Despool program,

**Type** /CCUTIL/DESPOOL

**Press**

The following menu appears:

```
Despooling from slot 5, server SERVER
Valid options are:

Start      - Start despooling
Device     - Despool device .....Printer
ExpTabs   - Expand tabs .....8
MaxLines  - Max lines/page .....58
PipeName  - Despool pipe name .....PRINTER

LineFeed  - Insert line feeds ..... YES
Header    - Print header page ..... YES
Trailer   - Print trailer page ..... YES
Exit      - Exit program

AltSlot   - Set alternate slot
Status    - Current pipe status
```

The despool options that must be set are the PipeName, and Device. The PipeName must be the same as the name when the file was spooled, in this case LETTER. To set the despool option Device,

**Press**

The following menu appears with corresponding function keys:

```
Valid despooling devices are:

Console - Output to console
File    - Output to file
Printer - Output to printer

Exit    - Exit
```

If there is no printer attached to the Concept, then the Printer option does not appear. To set the output to any of the options, press the appropriate key, and the main menu reappears.

**Press** **[File]**

The main menu is updated, and reappears. At this time the options should be ready to despool to a file.

**Press** **[Start]**

You are prompted for the destination file name in the Command window:

```
Spooled file name: /VOL3/TESTPROG.TEXT
Enter file name to DESPOOL to: /VOL3/TESTPROG.TEXT
```

The default file name is the same as the file name that was spooled. If you want to despool the file to a different volume or file, type the file name that you want and press **[RETURN]**. The file begins to despool immediately.

If the pipe is being despoiled to a file and the **[BREAK]** key is pressed, you are prompted for pipe status and file status:

```
Pipe status:
File: /VOL2/LETTER.TEXT
Pipe: DUMMY [1]

Despooling aborted

Valid pipe dispositions are:
ClosPipe - Close pipe
PurgPipe - Purge pipe
```

If the pipe is closed it remains in the Pipes area to be despoiled. Only the portion of the file that was not despoiled previously remains in the pipe, and at least one block was lost. Purging the pipe removes the remainder of the file in the pipe. After pressing either **[ClosPipe]** or **[PurgPipe]** the program prompts for the file disposition:

```
Valid file dispositions are:
ClosFile - Close file
PurgFile - Purge file
```

After responding to this prompt you are placed in the main menu of the despool program. At this point the file has been despoiled and you may Exit. Press **[Exit]** and you return to the level from which Despool was called.

## **VOLUME MANAGEMENT**

The CCOS can handle only 23 volumes at one time. Therefore only 23 can be mounted, or on-line, at once. There are two programs in the Dispatcher level that help manage volumes. They are the Constellation II program, and the Mount Manager program.

The Mount Manager allows you to control which volumes you have mounted, and the Constellation II function allows the system manager to manipulate volumes and user accounts. The Constellation II level is for the exclusive use of the system manager and is not covered in this guide. Its use is discussed in "The Corvus Concept System Manager's Guide."



If there are more than 23 volumes in your account, you may find that you cannot access them all. To be accessible, a volume must be mounted. The Mount Manager program allows you to mount and unmount volumes, thereby accessing more than the 23 available at boot.

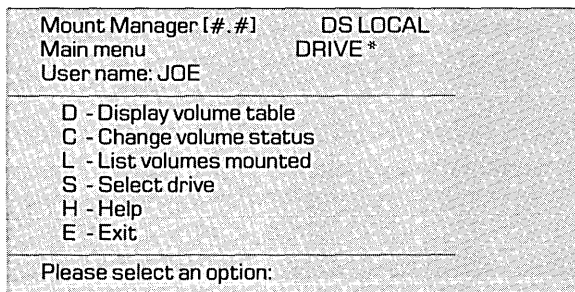
To run the Mount Manager program, use the MountMgr function in the Dispatcher level.

**Press** MountMgr

The following labels display:

<b>MOUNT MANAGER</b>									
<b>F1</b>	<b>F2</b>	<b>F3</b>	<b>F4</b>	<b>F5</b>	<b>F6</b>	<b>F7</b>	<b>F8</b>	<b>F9</b>	<b>F10</b>
Change			Display		List Vol		Select		
volume			volume		Mounted		drive	HELP	EXIT

In the System window the Mount Manager's main menu displays:



Each of the commands may be entered by either the appropriate function key, or by its corresponding letter as is shown in the table above. The function keys are available in either shifted or unshifted form.

The help command in the Mount Manager routine is very limited, and writes only small descriptions of the different commands. To look at the help commands,

**Press** `[HELP]`

or

**Press** H

The following message displays in the System window:

```
Mount Manager [#.#]      DS LOCAL
Help menu                DRIVE *
User name: SMGR

-----
Display Volumes - Display all volumes to
which user has access.

List Vol Mounted - List only the
volumes that are currently mounted.

Change Volume - Change a volume's
mount or read/write attributes.

Select Drive - Specify drive to be used
for Display, List, or Change.

-----
Press <space> to continue, or
press F to list to a file.
```

The bottom lines of the message are consistent throughout most of the Constellation programs. Pressing `[SPACE]` returns to the Mount Manager's main menu. This same line displays at the bottom of the Help, Display volume, and List Vol Mounted tables. The information in any of those tables can be sent to a file or to the printer by following these steps:

**Press** F

The program responds by prompting you with a default of /PRINTER:

```
Press <space> to continue, or
press F to list to a file.
Enter name of file: /PRINTER
```

A local printer must be attached with the printer driver assigned in order to send the table to a printer. The table may be sent to a file:

**Type** /VOL1/TABLE.TEXT

**Press** RETURN

The information in the menu, or table is located in the file TABLE.TEXT in VOL1, and it could be spooled to a printer so that you do not need to enter Mount Manager every time you want to see your mount table.

## Display Volumes Accessible

To display a list of the volumes available to you on a particular drive,

**Press** Display Volumes

The following table appears:

Mount Manager [#.#]		DS LOCAL	
Display volume table		DRIVE *	
User name: JOE			
Drive:	Drive1		
Volume	Unit	RW	TYPE
CCSYS	4	x	CCOS
VOL1	5	x	
CCUTIL	11	x	CCOS
VOL7	12	x	
VOL3	13	x	
VOL2	14	x	
VOL5	15	x	

In the Command window you are told to "Press <space> to continue." If you press **[SPACE]** the following message displays in the System window:

```
Number of volumes accessible: 7
Press <space> to continue, or
press F to list to a file.
```

## Display Volumes Mounted

To display a list of the volumes mounted for you on a particular drive,

**Press** **[List Vol Mounted]**

The following table appears:

Mount Manager [#.#]	DS LOCAL		
List volumes mounted	DRIVE *		
User name: JOE			
Volume	Unit	RW	Type
CCSYS	4	x	CCOS
VOL1	5	x	
*	9	x	
CCUTIL	11	x	CCOS
VOL3	13	x	
VOL2	14	x	
VOL5	15	x	
Number of volumes mounted: 7			
Press <space> to continue, or press F to list to a file.			

## Selecting a Drive

If the system has more than one drive attached, and you want to look at the volumes on a drive other than the boot drive, you must use Select Drive. To do this,

**Press** `[Select Drive]`

The following appears:

```
Mount Manager [#.#]    DS LOCAL
Select drive          DRIVE *
User name: JOE
-----
Enter drive information:

Slot 2 contains a Corvus disk interface
Slot 5 contains an OMNINET interface

Slot number {2,5}: 2
```

If the drive is a local drive press `[RETURN]`, but if the desired drive is on Omninet press 5 and `[RETURN]`.

**Press** `[RETURN]`

The Concept answers the next prompt itself and continues as follows:

```
Slot number {2,5}: 2
Server name: LOCAL
Valid drives are: DRIVE1,
Drive name: *
```

Both of the first two prompts are automatically answered by the program and the third defaults to \*, the current drive. If you do not want to access the current drive, type the name of the desired drive and press `[RETURN]`.

## Changing Volume Attributes

To make any change of a volume's attributes enter the Mount Manager and then,

**Press** `[Change volume]`

All changes to a volume can be made in this program. After pressing `[Change volume]` the following displays in the System window:

```
Enter volume name: _____
```

Type the name of a volume. In this example volume VOL7 is going to be mounted.

**Type** VOL7

**Press** `[RETURN]`

The program continues:

```
Mount Manager [#.#]      DS LOCAL
Change volume table     DRIVE *
User name: JOE

Current volume attributes:

  Volume name: VOL7
  Mount Status: Unmounted
  Mount Unit: *
  Access: RW

Change to:
  Volume name: VOL7
  Mount status: Unmounted
```

To keep this volume unmounted, simply press `[RETURN]`, and accept the default. To mount the volume,

**Type** Mounted

**Press** `[RETURN]`

The next prompt line appears:

```
Unit number (5,9..30): 12
```

This prompt asks for the logical unit number that the volume should be mounted on. There is a default value that appears on the screen which is the first available number that is greater than nine. If you wanted to change its unit number to seventeen,

**Press 17**

**Press** `RETURN`

The last prompt appears next:

```
Access (RO/RW): RW
```

The default is read-write, its current status. If the volume has read-only status, you are not given the option of changing it to read-write. That change must be made in the Constellation II program by the system manager.

If you have some reason to change a volume from read-write access to read-only you can do that here. The change is reversible, since the change would not affect the Constellation II tables.

## Exiting the Mount Manager

To exit the program press **[Exit]**. Before being returned to the Dispatcher level a prompt appears asking if these changes should be made permanently or not. If not, these changes remain in effect until the Concept is rebooted or powered off.

Do you wish to save the changes permanently? Y

Pressing **[RETURN]** changes the boot tables and saves the changes permanently. To make these changes temporary,

**Press N**

**Press [RETURN]**

The screen clears and the appropriate level (Dispatcher or File Manager) returns.

## THE VOLUME UTILITIES PROGRAM

The program /CCUTIL/VOLUTIL provides simple methods for backing up volumes onto diskette, and copying diskettes. VOLUTIL also shows the Concept's system status, giving a brief summary of its configuration.

To execute the volume utility program,

**Type /CCUTIL/VOLUTIL**

**Press [RETURN]**



A menu appears on the screen with corresponding function keys in the label window:

```

Valid options are:

[BckupVol] (B) - Backup volume to floppy disk
[CopyFlpy] (C) - Copy floppy disk

[SysmStat] (S) - System status

[Exit]      (E) - Exit program
    
```

### VOLUTIL

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
									Exit
			BckupVol	CopyFlpy	SysmStat				

**[BckupVol]**

is the volume backup program. It allows you to backup a volume onto diskettes.

**[CopyFlpy]**

is the floppy diskette copying program. It allows you to write a floppy diskette to the hard disk, and then make copies of the diskette.

**[Exit]**

leaves the program. It returns to the level of the CCOS from which the program was called.

**[SysmStat]**

gives the system status report. It tells how much memory is available in RAM, what types of drives are on-line, and much more.

## Backing Up Volumes

The first function in VOLUTIL is the BckupVol program. This program takes a volume from a hard disk drive, and transfers it to floppy diskettes.

There is no restore function to this program. If you have a file that has to go on multiple floppies, it has to be restored manually. This is done with the FileMgr CopyFile and ConcFile functions, and is discussed later. To run this program,

**Press** `[BckupVol]`

The backup program does not include options or other functions. It asks for the volume's name, which diskette drive to copy to, and then gives instructions for switching diskettes as it copies the information.

After pressing `[BckupVol]` the screen clears and the first question is asked:

A screenshot of a terminal window with a dark background and light-colored text. The text reads "Backup which volume?". The prompt is enclosed in a rectangular box with a light gray background and a dark border.

**Type** VOL3

**Press** `[RETURN]`

The program does not expect a slash for a volume name delimiter, since it is asking for a volume name.

The program looks at the volume and prints the number of files and allocated blocks it has. It must look at this for its own use to determine how many floppy diskettes have to be used to make the copy.

```
Volume VOL3 contains 12 files (724 allocated blocks)
Backup to which floppy disk unit? 9
```

The volume VOL3 contains 724 allocated blocks. This is the number of blocks that are used, and does not include the empty space in a volume. The actual size of VOL3 might be 1024 blocks, 2048 blocks, or any size capable of holding those files.

The question asks to which floppy disk unit number to copy the information. The default is unit 9, since that is the number given the first diskette drive attached to the Concept. This number is to be changed only if there are two or more diskette drives attached to the Concept and the first one is not to be used. The second drive usually has the unit number 10.

Before accepting this value by pressing **RETURN**, place a diskette in the drive. The program does not recognize the drive if it does not have a diskette ready.

**Press** **RETURN**

The program gives an instruction here to place the destination diskette in the drive.

```
Insert floppy disk (1 of 2) in unit 9, ready? [Y/N]:
```

If the diskette that was placed in the drive is not the same diskette that the information is to be backed up on, change it now. Do not be alarmed if you accidentally pressed **[RETURN]** at this point with the wrong diskette in the drive. The program checks the diskette for any files.

**Press** **[RETURN]**

If the diskette has any files on it the following message is given, and a question is asked:

```
Volume floppy contains 3 files
OK to delete all files on volume FLOPPY? (Y/N):
```

If this is the proper diskette,

**Press** **Y**

**Press** **[RETURN]**

The copy process starts at this time, and after each file has been moved a message like the following appears:

```
Volume VOL3 file 1ST.PRPOSL.TEXT backed up
( 17 blocks)
```

After a short period of time, the diskette becomes full. The program stops, and gives another instruction in the System window and Command window:

```
Remove floppy disk (1 of 2) from unit 9
```

```
Press <space> to continue
```

Remove the first diskette, and

**Press** **[SPACE]**

The procedures are the same for the second and subsequent diskettes as for the first. After pressing **[SPACE]** the program starts with "Insert floppy disk (2 of 2) ..." and finishes the copy. After this procedure is completed the diskettes have the names BACKUP1, BACKUP2, and so on.

To leave this program after the copy is complete press **[SPACE]** to the "Press <space> to continue" instruction. To leave it before completing the copy press **[BREAK]**. This terminates the task, but does not provide a useable copy of the diskette that is being processed at the time.

## **Restoring a Volume**

To restore a volume that is on more than one diskette, follow the procedure that is given here.

Place the first diskette in the diskette drive.

**Press** **[FileMgr]**

**Press** **[CopyFile]**

All the files from this diskette need to be copied onto the new destination volume. To do this,

**Type** /BACKUP1/\*

**Press** **[RETURN]**

The Copy File program now asks for the destination volume name. After the destination volume name add the quiet option for an easier copy.

**Type** /VOL6 -Q

**Press**

The files are automatically copied into the volume VOL6. Repeat this process for the second diskette, BACKUP2.

Quite often there will be a file that spans the two diskettes. Each part of the file is given a unique name. If the file is called LABELS.TEXT the first name becomes "1LABELS.TEXT" and "2LABELS.TEXT" is the name for the second half of the file. To combine the file,

**Press**

The following prompt appears:

Concatenate which files?

**Type**

/VOL6/1LABELS.TEXT /VOL6/2LABELS.TEXT

**Press**

The following prompt appears:

Output concatenated files to where?

**Type** /VOL6/LABELS.TEXT

**Press**

The files are merged and the output is placed in the original file, and in the appropriate volume. After listing the volume to be sure the file exists, remember to delete the two partial files.

## Copying Diskettes

The second option in the VOLUTIL program is Copy Floppy. Copy Floppy takes the information from the diskette and copies it into a file large enough to hold all the information on the floppy. If the floppy holds 500 blocks, that is the length of the file that will be created on the hard disk. The image that is on the hard disk can be copied back onto diskettes until the file is deleted.

To run the program,

Press **CopyFlpy**

A new menu displays and the function keys change. They look like the ones following:

```
Valid options are:

[ReadFlpy] - Read floppy disk to work file
[SetIntlv]  - Set floppy disk interleave values

[WritFlpy] - Write from work file to floppy disk
[SetWorkF] - Set work file name
[SetFlpyU] - Set floppy unit number
[DelWorkF] - Delete work file

[Exit]     - Exit copy floppy processing
```

COPY FLOPPY						
F1	F3	F5	F6	F7	F8	F10
ReadFlpy	SetIntlv	WritFlpy	SetWorkF	SetFlpyU	DelWorkF	Exit

**DelWorkF**

deletes the work file used by the program to store the diskette's information on the hard disk.

**Exit**

leaves the CopyFlpy program returning the system to the VOLUTIL main menu.

`ReadFlpy`

copies the information from the diskette to the hard disk. The work file name is the destination file for the read floppy function.

`SetFlpyU`

sets the unit number of the diskette drive that is communicating with the program. The default is unit 9. With only one floppy diskette drive attached this number will never need to be changed.

`SetIntlv`

sets the values of the interleave for the floppy disk drive to handle different formats of diskettes. The program prompts for the new values, always defaulting to the settings normally used by Concept diskettes and software. The values should never be changed unless the exact interleave values of the diskette unit are known.

`SetWorkF`

sets the work file on the hard disk that is used by the program to transfer the diskette's data from one diskette to another. The file defaults to the file name ...FLOPPY.DATA in the current volume. The volume must have 500 blocks available in order to use this program. More space is needed for larger floppy disk volumes.

`WritFlpy`

copies the information from the hard drive to the diskette. It is used for creating the duplicate after the original file has already been written to the disk using the ReadFlpy function.

This program is used to make one or more copies of a diskette. The procedure for doing this is to set the work file name, then read the diskette, and finally write the information back to diskettes.

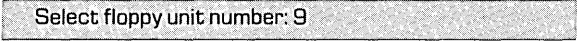


**Setting the Floppy Unit Number.** This option is used only when there is a second diskette drive connected to the Concept, and that drive is being used for the CopyFlpy function instead of the first. The diskette drive's logical unit number is defaulted to 9 by the program. The first diskette drive attached to a Concept has a logical unit number of 9.

To change the floppy unit number,

**Press**

The following prompt displays in the System window:



Select floppy unit number: 9

**Type** 10

**Press**

The floppy unit number is changed and the CopyFlpy menu reappears.

**Setting the Work File.** This option sets the name of the file that is created for the copy. If the file is going to be saved and used over and over, the file should have a descriptive name. That is why this is a far more commonly used option than SetFlpyU.

The work file contains the data that is needed as long as copies are being made. The work file must be in a volume that has 500 blocks of available space (more if the floppy volume is more than 500 blocks).

Since the default work file name is ...FLOPPY.DATA it is sent to the current volume. If the current volume does not have enough free blocks an error message displays.

Find a volume that has enough free blocks, VOL7 for this example, and set the work file name to a file in that volume.

**Press** `SetWorkF`

```
Enter new work file name: ...FLOPPY.DATA
```

The default of ...FLOPPY.DATA appears on the screen, but any file name can be entered.

**Type** /VOL7/MASTERFLPY

**Press** `RETURN`

The work file name is set and the CopyFlpy main menu reappears.

**Reading a Diskette.** Reading a floppy diskette is the first step in actually copying a diskette.

ReadFlpy copies the information from the diskette to the file specified as the work file. To begin the copy,

**Press** `ReadFile`

The following displays on the screen, although the numbers on the right are constantly changing as the copy progresses:

```
Source file: Floppy unit  9          8
Destination file: /VOL7/MASTERFLPY 7 <
```

Each time a block is copied the numbers increment by one. As soon as the numbers reach 500 the floppy has been copied to the appropriate file and the CopyFlpy menu reappears (the number must reach 1001 or 2002 for one- or two-thousand block volumes).

If this function has been performed already and the work file has not been deleted a prompt appears right after pressing **[ReadFlpy]**:

```
File already exists, delete it? [Y/N]:
```

The file mentioned is the old work file, and if you answer yes, only that file is destroyed. The floppy diskette is not damaged or affected in any way. To continue from this point press Y and the program continues. Pressing N stops the program, so that you may set a new work file name.

**Writing a Diskette.** Writing a diskette is very similar to reading the diskette. Before a diskette can be written, the ReadFlpy function must have been performed to create a work file full of data.

WritFlpy sends the information from the work file to the diskette in the drive. There must be a work file available to copy, and the work file name must be the same as it was when it was written. The file name does not have to be changed unless CopyFlpy was exited, otherwise it remains the same. To begin the copy,

**Press** **[WritFlpy]**

There is one question to answer, and it appears every time a copy is made.

Destroy data on floppy disk? [Y/N]:

This question is asked regardless of whether there are actually any files on the disk. Do not be overly concerned when this appears after putting in a diskette that was supposed to be new or blank. It probably is blank. But on the same token, do not put in a diskette that has any information that is going to be needed later, because it will be destroyed.

### **Press Y**

The copy process begins, and the screen display is the same as the ReadFlpy, with one exception: the source file is /VOL7/MASTERFLPY and the destination file is Floppy unit 9.

This function does not destroy the work file, so WritFlpy can be performed over and over to create as many floppy diskette copies as is desired.

As soon as the copy is complete the CopyFlpy menu reappears.

**Deleting the Work File.** There is a key that deletes the work file automatically, so that the volume space can be regained without going out of the program to the File Manager. To delete the work file,

**Press**

A message appears and the main menu stays in place:

```
Workfile /VOL7/MASTERFLPY deleted ...
```

## **System Status Information**

This program gives information regarding the hardware and other internal parts of the Concept. There is information on: the slot assignments, to tell what devices are attached to the Concept; Corvus disk parameters, to tell what types of drives are attached, what size they are, the ROM, and the type of firmware on the disk; system parameters to tell about the size of memory in the machine, the stack pointer position, the Boot PROMs in the machine, and more; the amount of code and stack memory available given both in numbers and through a graphic display.

# ADVANCED FEATURES | 7

This chapter discusses a variety of advanced features that are quite useful. Executable files, startup files, user definable function keys, data communications, and the terminal program are among the features discussed.

## USING EXEC FILES

An Exec file is a text file that contains a list of CCOS commands and program names which you want to execute in a particular order. You can use the Concept command interpreter to execute, in sequence from top to bottom, the contents of the list.

Valid entries in an Exec file are the file name of any program, and the commands described in "The Corvus Concept Operating System Reference Manual."

In the following example, several programs are listed in a text file named DCHECK.TEXT in the current volume /VOL1. One program is the Corvus diskette formatting program, one is the Corvus diskette diagnostic program, one is the Corvus disk backup program, and the last one is the Corvus disk diagnostic program. Each program name must be on a separate line of the file.

```
/CCUTIL/FFRMT  
/CCUTIL/FDIAG  
/CCUTIL/MIRROR  
/CCUTIL/DDIAG
```

ExecFile runs the contents of an ExecFile. To execute the contents of this file from the Dispatcher level,

**Press** `[ExecFile]`

The Command window displays:

```
Execute which command file?
```

**Type** `/VOL1/DCHECK.TEXT`

**Press** `[RETURN]`

The programs listed in the file are run in the order in which they are listed.

Exec files come in handy when you compile or link a program often. The next example shows how to set up an Exec file to run the linker for a specific program. This saves you the effort of typing several file names every time you want to link the program. The file being linked is called MAIL.OBJ, which is linked to a unit called SETBYTE.OBJ and three libraries.

```
LINKER MAIL SETBYTE /CCUTIL/C2LIB /CCUTIL/CCLIB IPASLIB
```

Do not include `/CCSYS/ED` (EdWord), or any EdWord commands in the middle of an Exec file. This may cause problems with the workspace you try to enter. In order to run EdWord from a command file be sure that it is the last line of the file. A 512K Concept is required. The workspace name can be specified as follows:

```
!ED /VOL2/W
```

## Automatic Startup File

Using an Exec file, CCOS commands and programs can be run automatically when the Concept is booted. To do this you create an Exec file called STARTUP.TEXT and place it in either the home volume or the boot volume.

When the Concept is booted, the CCOS searches the boot volume and the home volume, mounted on unit number 5, for the file STARTUP.TEXT. If this file exists in the boot volume or in your home volume, it is used as a source of commands to the operating system.

If the file is in the boot volume it is used as a system wide startup file. Everyone that uses the system has the file executed at boot. If the file is in the home volume then it is unique to the individual, or individuals using that volume as their home volume.

If there is a startup file in both the boot volume and the home volume, they are both executed. The system wide startup file is performed first, then the personal file.

The following is a useful example of a startup file in the volume mounted on unit number 5:

```
!CC.WNDMGR CSDISP /CCUTIL/CSD.07.11
!CC.SETDCP UNIT=PRINTER
!CC.FILMGR SETVOL /VOL2
!ED W
```

The exclamation point used at the beginning of each of these lines tells the Concept to look at the system volume; it is an abbreviation for the boot volume.



The first line of the file calls the Window Manager program to load a display character set. The character set to be loaded is /CCUTIL/CSD.07.11, the standard character set for use with character enhancements in EdWord.

The second line calls the Set Data Communication parameter program. That is where the printer driver is assigned.

The third line calls the File Manager routine for setting the current volume.

The last line executes EdWord the Editor/Word processor program, and enters the workspace W in the current volume.

These command lines may appear confusing, but if any of these capabilities are appealing you can type the lines as they appear above into a file named STARTUP.TEXT and use them. The only variables in the lines above are the display character set (/CCUTIL/CSD.07.11), the volume name to be set (/VOL2), and the EdWord workspace name (W).

For more information on what can be done at the Exec level, see "The Corvus Concept Operating System Reference Manual."

## DEFINING USERKEYS

There is a file called USERKEYS.TEXT that allows you to create your own levels of function keys. The system looks for the file USERKEYS.TEXT in the volume mounted on unit 5 at boot, and if it is present, Userkeys appears as a label in the Dispatcher level.

USERKEYS.TEXT is a text file, and can be edited using EdWord. At the top of the text file are comment lines which help to instruct you in creating your own labels. There are already labels present in the file, but they can be changed by simply typing over them.

A single userkey is defined by two or more lines in the following manner:

```
(level) (keynumber) (keylabel)
(command)
[optional (command)]

1 0 Pascal
!PASCAL

1 2 EdWord
!ED

1 3 7.11.ALT
!CC.WNDMGR CSDISP /CCUTIL/CSD.07.11.ALT
```

The exclamation mark before PASCAL or ED in the example tells the computer that the file is in the system volume. The same information could be entered as /CCSYS/ED instead of !ED.

The userkey defined after EdWord loads the seven by eleven alternate character set. It calls the Window Manager and then runs the load display character set program. The character set that is loaded is the default character set used by EdWord.

The level of Userkeys is from one to ten. Level one is the first level of labels that appear when Userkeys is pressed. If from the first level NxtLevel is pressed the second level appears, etc., until the tenth level. If there are only three levels defined, only three levels appear. All ten exist only if at least one key is defined with the level ten.

The keynumber defines which function key is to be used. The following list shows the correspondence between the numbers and the key position:

- 0 - 9 keys F1 - F10
- 10 - 19 keys F1 - F10 with shift
- 20 - 29 keys F1 - F10 with command
- 30 - 39 keys F1 - F10 with command and shift

The keynumbers 9, 19, and 29 are reserved for use by the Dispatcher.

The keylabel is the name that appears in the Label window. The keylabel must be eight characters or less.

Command is the program name that is to be executed when the key is pressed. It should specify the volume name enclosed in slashes followed by the program name, i.e. /VOL1/MYKEY.

After editing the file so that the function keys appear the way you want them to, execute the program CC.DISPAT in the volume /CCSYS. The program reinitializes the Dispatcher level and includes the new Userkeys, so long as the file is in the home volume.

Check to make sure the keys are correct. Once USERKEYS.TEXT has been placed in the home volume, Userkeys are available to you every time the system is booted.

Normally, after booting the system the Dispatcher level function key labels display. Userkeys allows the default level to be changed. If the keynumber of 40 is found in the first line of the file USERKEYS.TEXT, then the Userkeys level specified in the first line displays at boot. For example, if the second Userkeys level is to be the default, the first line of the text file would look like this:

```
2 40
1 0 Pascal
!PASCAL
1 2 EdWord
!ED
...
```

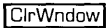
The Concept boots to the second Userkey level.


There are two other functions in the Userkey command file. They are for performing startup operations. If the level is 98 and the keynumber is 98 the program processes one command the first time the Dispatcher level appears at boot. If the level is 99 and the keynumber is 99 the program executes one command each time the Dispatcher level is executed. An example of the form for these commands is as follows:


```
...  
98 98  
!ED /VOL2/W  
  
99 99  
/VOL3/SCHEDULE
```

The user would enter EdWord automatically at boot every time he logs on because of the command following "98 98." Every time the user returns to the Dispatcher level a program called SCHEDULE in volume /VOL3 is run because of the command following "99 99."

Three commands are recognized by keywords in the system. They are:

CLRWND - same as pressing the  key.

REBOOT - same as pressing the  key.

RESTART - same as pressing the  key.

They are used in a file as follows:

```
1 5 ClrWindow  
CLRWND  
  
1 7 Restart  
RESTART  
  
1 39 Reboot  
REBOOT
```

## USING THE RUNTIME UCSD P-SYSTEM

To use the runtime UCSD p-System a few steps must be performed to ensure that the system will run. The p-System key appears in the Dispatcher level, but it is inoperative unless the following conditions are met:

1. The current volume must be a UCSD volume. To change a CCOS volume to a UCSD volume use the ChngVol function in the FileMgr level. Never run the p-System with /CCSYS as the current volume.
2. The file CC.PSYSTEM must be in the boot volume /CCSYS. The file should already be in /CCSYS, which is where the file is placed when the disk is initialized.
3. The files SYSTEM.PASCAL, SYSTEM.LIBRARY, and SYSTEM.MISCINFO must be in the current volume. These files can be copied from the volume /CCUTIL.
4. The file SYSTEM.INTERP must be in either the current volume or the system volume, /CCSYS. The file should already be in /CCSYS, which is where the file is placed when the disk is initialized.

If all of these conditions are met the runtime UCSD p-System can be used. To enter the p-System,

**Press** p-System

All UCSD volumes you have access to are displayed, and if there are less than six they are mounted automatically. If there are six or more UCSD volumes the p-System asks if each volume should be mounted. A maximum of five optional volumes can be mounted in the p-System. The current volume is always mounted. The p-System considers the RAM disk to be a volume, so it is included as one of the five volumes from which you may choose.

A few other details to keep in mind include: interrupting screen display, printing, window size, and character type. The **CTRL**-S and **CTRL**-Q commands are still used for stopping and starting screen display, respectively. To use a printer from the p-System the printer driver must be assigned in the CCOS before entering the p-System. Different character sets and different window sizes may be used in the p-System, but they too must be set in the CCOS before entering the p-System.

In order to leave the p-System and return to the CCOS press H for Halt; there is no exit key.

## **SET DATA COMMUNICATION PARAMETERS**

SetDtaCm loads the DataCom drivers, and allows you to set the parameters for the type of printer to be used. This enables you to communicate with modems, terminals, or other devices through the RS-232C ports.

The printer driver must be assigned before a printer can be used. The procedure described below enables you to do this.

**Press**

**Press**

After pressing  the labels appear as shown below:

UNIT TYPE									
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Printer		DataCom1		DataCom2					NoChange

If you select Printer the printer driver is loaded. If DataCom1 is pressed the DataCom driver is loaded, and the communication parameters for the device connected to Port 1 can be set. If DataCom2 is pressed the DataCom driver is loaded, and the communication parameters for the device connected to Port 2 can be set. NoChange exits the program without assigning a driver.

To load the printer driver,

**Press**

This loads the printer driver and displays the default printer values:

Baud rate: 1200 Parity: Space Xmit/Ignore RcvPort: 2 Char size: 7 bits Protocol: Xon/Xoff Unit: Printer
--



The default parameters for the two data communication ports are set so that DataCom1 is useful with a terminal and DataCom2 is useable with a NEC Spinwriter printer. The default settings for the DataCom ports are as follows if they are selected. The settings for DataCom2 change when the printer driver is loaded.

#### DataCom1:

Baud rate: 9600 Parity: Disabled Port: 2  
Char size: 8 bits Protocol: Xon/Xoff Unit: DataComm1

#### DataCom2:

Baud rate: 4800 Parity: Disabled  
Char size: 8 bits Protocol: Line/DSR/Normal Unit: DataComm2

After selecting the unit the labels change, and appear as they do here:

### SET DATA COMMUNICATIONS

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
BaudRate	Parity	Port	CharSize	Protocol	Unit	PrtFunc	TestMode	PrtType	Exit

**BaudRate**

is used to set the baud rate.

**CharSize**

is used to set character size.

**Exit**

returns you to the SysUtils level.

**Parity**

is used to set the parity.

**Port**

is used to select which RS-232C port is being setup.

**Protocol**

is used to set handshake protocol.

**PrtrFunc**

is used to set the number of lines per inch, characters per inch, auto line feed ON/OFF, and load printer action tables or alternate character translation tables.

**PrtrType**

is used to set all the parameters according to standard defined printer types.

**TestMode**

is used to test the parameters by sending or receiving data character by character.

**Unit**

is used to select the driver that is to be used. It is either the PRINTER driver, or the DTACOM driver associated with one of the RS-232C ports.

The Port, PrtrFunc, and PrtrType labels display only when the unit Printer is selected.

In the following descriptions of the SetDtaCm labels, each group of labels includes a NoChange option. Pressing **NoChange** returns you to the SetDtaCm labels without making any changes.

PrtrType is used to set all the SetDtaCm values for three specific types of printer: a NEC 7710, an Epson MX-100, and an IDS 460.

**Press** **PrtrType**

The PrtrType labels display:

**PRINTER TYPE**

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Epson	NEC	IDS							NoChange

Press the appropriate key to set the parameters for the type of printer to be used. The default values for these printers are as follows:

### EPSON —

Baud rate: 4800 Parity: Disabled Port: 2  
Char size: 8 bits Protocol: Line/DSR/Normal Unit: Printer

### NEC —

Baud rate: 1200 Parity: Space Xmit/Ignore Rcv Port: 2  
Char size: 7 bits Protocol: Xon/Xoff Unit: Printer

### IDS —

Baud rate: 9600 Parity: Disabled Port: 2  
Char size: 8 bits Protocol: XON/XOFF Unit: Printer

### BaudRate is used to set the baud rate

To change the baud rate, press **[BaudRate]**, and the following labels display.

Select the key that corresponds to the baud rate on your printer or other device. The Set Data Comm level automatically returns when any key is pressed.

#### BAUD RATE

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
300	600	1200	2400	4800	9600	19200			NoChange

**Parity is used to set parity.** To change parity, press **[Parity]**, and the following labels display.

Select the key that corresponds to the parity on your printer or other device. The Set Data Comm level automatically returns when any key is pressed.

**PARITY**

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Disable	Odd	Even	MrkXNRc	SpCXNRc					NoChange

Port is used to choose which Concept RS-232 port is used for communications. To change the output port, press **[Port]**, and the following labels display.

Port 2 is closest to the Omninet port on the back of your Concept. Select the key that corresponds to the RS-232 port to be used. The Set Data Comm level automatically returns when any key is pressed.

**PORT**

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Port1	Port2								NoChange

CharSize is used to change transmission character size. To change the character size, press **[CharSize]**, and the following labels display.

Select the key that corresponds to the character size of your printer or other device. The Set Data Comm level automatically returns when any key is pressed. The settings for the CPI are only displayed if the printer selected is IDS.

### CHARACTER SIZE

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Epson	NEC				IDS				NoChange

Protocol is used to change the handshake protocol. To change the handshake protocol, press **[Protocol]**, and the following labels display.

### PROTOCOL

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Line	Xon/Xoff	Enq/Ack	Ext/Ack	NoProto					NoChange

Select the key that corresponds to the protocol on your printer or other device. The Set Data Comm level automatically returns when any key is pressed.

Printer functions is used to set a number of printer parameters. The number of characters per inch and lines per inch may both be set here. The default value of each displays on the screen. Characters per inch defaults to 10, and lines per inch defaults to 6.

Another parameter that displays in the Command window is the setting for automatic line feeds (LF). If the printer has a default of LF on, the Concept's setting should be off, and vice versa. The settings should be opposite each other.

The commands LdAction and LdAltChr are used to load a printer action table or an alternate character translation table.

A printer action table contains the character sequences for a given printer necessary to perform certain operations to create enhancements (such as bold face).

An alternate character translation table contains the character sequences for a given printer necessary to create the alternate characters. The characters are the upper 127 characters of the alternate character sets.

These tables may be created using the programs BLDACT and BLDALT found in /CCUTIL. Instructions for using these programs can be found in the "The Corvus Concept Operating System Reference Manual."

The distribution diskette /FSYSGEN contains printer action tables for the CITO H printer and the Diablo 630. These files begin with the prefix PAT. The default settings for the printer action table are those of the NEC 7710.

Pressing  returns you to the Set Data Comm level.

The Command window and the Label window appear as follows:

```
Characters per inch (CPI): 10  Lines per inch (LPI): 6  
Auto Line Feed (LF): ON
```

## PRINTER FUNCTIONS

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
10 CPI	12 CPI	6 LPI	8 LPI	LdAction	LdAltChr	LF on	LF off		Exit

The test mode is used to test the communication parameter settings by sending or receiving data. Pressing **[TestMode]** prompts for the type of test to run. The Command and Label windows appear as follows:

Test mode, Send or Receive :

## TEST MODE

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Send	Receive								NoChange

**Press** **[Send]**

The System window prompts for the length of the test sample:

Enter number of characters to send:

Type in the number of characters that you want to try as a test of the data communication compatibility.

**Type** 10

**Press** **[RETURN]**

Another prompt appears:

Enter number of characters to send: 10  
Enter characters to send:

Now type in any ten characters you choose (type more if the number you entered just above is greater). If you are testing the send on a printer, the printer should act as a typewriter. Many serial printers require that you press `[RETURN]` before any characters are printed. The characters typed should be sent to the printer, or other device, without problems.

If the printer parameters are not set correctly you may receive either garbage, or nothing at all depending on the problem. If it is garbage, check the parity and character size for compatibility. If nothing at all is received check the baud rate settings.

The receive test mode works in essentially the same way except that the Concept is on the receiving end of the transmission. You must be hooked up to a device other than a printer, since a printer is made for one way communication only and cannot send data to the Concept.

## **ASSIGNING A DRIVER**

A driver is a system program that controls devices, such as printers, CRTs, disk drives, diskette drives, and volumes. It is the driver that makes the different parts of the Workstation interact with each other.

`AsgnDrvr`, in the System Utilities level of labels, allows you to assign a driver to a device. This function is primarily of use to those who are developing their own drivers.



The following example shows you how to assign the Concept printer driver to a printer using AsgnDrv.

The printer driver could be assigned by pressing the Printer key in the Set Data Comm program, but in this example you will see how a driver is manually loaded.

The printer driver must have the DataCom drivers assigned before it can be loaded. Therefore before trying this example make sure the printer driver has not already been assigned. If it has been set, reboot. Enter the SetDtaCm program and select unit DataCm2, then exit the program and do the following procedure.

**Press**

The Command window displays:

**Type** /CCSYS/DRV.EPRNT

**Press**

The Command window displays:

**Type** PRINTER

**Press**

The Command window displays:

**Press 6**

**Press** `[RETURN]`

If you enter the Set Data Comm program again, the select unit prompt does not come up, and the printer driver is loaded.

For more detailed information on assigning a driver, see "The Corvus Concept Operating System Reference Manual."

## List Drivers On-line

Drvrvrsn allows you to list all assigned drivers. To list the drivers enter the System Utilities level and press `[Drvrvrsn]`.

```
Select SysUtils function:
```

**Press** `[Drvrvrsn]`

The following list appears on the screen:

```
Unit 0: /NULL      unblocked resident 7-Jul-82 NULL device driver
Unit 1: /CONSOLE  unblocked loadable 7-Jul-82 SYSTEM driver
Unit 2: /SYSTEM   unblocked loadable 7-Jul-82 SYSTEM driver
Unit 4: /CCSYS    blocked resident 7-Jul-82 Local disk driver
Unit 5: /MYSTUFF  blocked resident 7-Jul-82 Local disk driver
Unit 6: /PRINTER  unblocked loadable 14-Jan-83 PRINTER driver (v 1.0)
Unit 9: /FLOPPY   blocked loadable 7-Jul-82 8" Corvus floppy disk driver
Unit 10: /MYVOL1  blocked resident 7-Jul-82 Local disk driver
Unit 11: /MYVOL2  blocked resident 7-Jul-82 Local disk driver
Unit 12: /MYVOL3  blocked resident 7-Jul-82 Local disk driver
Unit 28: /MYVOL4  blocked resident 7-Jul-82 Local disk driver
Unit 29: /MYVOL5  blocked resident 7-Jul-82 Local disk driver
Unit 30: /SLOTIO  unblocked resident 7-Sep-82 Corvus disk command driver
Unit 31: /DTACOM1 unblocked loadable 14-Jan-83 DATACOM driver (v 5.0)
Unit 32: /DTACOM2 unblocked loadable 14-Jan-83 DATACOM driver (v 5.0)
Unit 33: /OMNINET unblocked resident 7-Jul-82 NULL device driver
Unit 34: /TIMER   unblocked loadable 22-Oct-82 TIMER driver
Unit 35: /KYBD    unblocked loadable 10-Sep-82 KEYBOARD driver
Unit 36: /DISPLAY unblocked loadable 18-Nov-8 DISPLAY driver
```

To understand this listing it is best to break it down one section at a time. Using the sixth row as an example the different columns are analyzed below:

```
Unit 6: /PRINTER  unblocked loadable 14-Jan-83 PRINTER driver (v1.0)
```

- Unit 6: —            this is the logical unit number that is assigned to the device listed on this line.
- /PRINTER —        this is the name of the device. In this case it is PRINTER.
- unblocked —        this tells us that this device is not blocked. A blocked device is capable of storing files, an unblocked device is not.
- loadable —         this driver is loadable, and is not a resident driver. It must be loaded in order to be used. A resident driver is part of the operating system code.
- 14-Jan-83 —        this is the date that the driver was released by Corvus. The date helps to differentiate the driver from other versions of the same driver.
- PRINTER driver  
(v 1.0) —           this is the name of the driver. As you can see by the listing there are many different drivers, each of which control different devices.

## RUNNING THE TERMINAL PROGRAM

The terminal program allows a Concept to speak with other computers and act as a terminal. The DataComm driver must be set for the Concept to communicate before this program is run.

There are a few options which may be specified when calling the terminal program. Those options include the DataCom port that is being used to connect the Concept and the other computer, and whether the automatic linefeed should be on or off for both the sending of characters to other computers, and the displaying of received characters.

The terminal program is run by typing /CCUTIL/TERMINAL and pressing **[RETURN]**. None of the options have been specified, and the first prompt that appears is the following:

```
*** No DataCom port specified on command line
    For DATACOM 1 use DC1, for DATACOM 2 use DC2
Select DataCom port (DC1/DC2): DC1
```

To set the DataCom port to port 1 before entering the terminal program,

**Type** /CCUTIL/TERMINAL UNIT = DC1  
**Press** **[RETURN]**

The prompt shown above does not appear, and the terminal program is set to communicate with whatever device is connected to DataCom 1. In order to communicate accurately with the other computer the auto linefeed settings should be designated. These settings are made in the command line. The format for doing this is:

```
/CCUTIL/TERMINAL UNIT = DC 1
SENDALF = OFF RCVALF = OFF
```

Each of the Auto Linefeed parameters is optional and default to ON if not specified. The unit number is mandatory. It is prompted for if it is not on the command line.

Send Auto Linefeed ON means that the Concept will send a Linefeed after a carriage return, OFF will not. Receive Auto Linefeed ON means that the other computer will not send a Linefeed after a carriage return, OFF means it will.

The function keys in the terminal program appear as follows:

### TERMINAL PROGRAM

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
									Exit
RecvFile		SendFile		Printer	ClrWindow				

The Command window shows the following prompt:

```
Output to screen only
Enter text or select function:
```

Characters received from the Concept DataComm port can be saved in a text file or sent to a printer. To save the output to a file,

**Press** `[RCV File]`

The Command window displays the following:

```
Enter file name: ... ECHO.TEXT
```

To accept the file name ... ECHO.TEXT press `[RETURN]`. To change the name type the desired name and press `[RETURN]`. A file is created at this point even if the file is not used. The `[RecvFile]` key turns the output off and on: by pressing the key again, the output is not sent to a file, only the screen, and the file is closed. The same is true for the `[Printer]` key. To send the output to a printer as well,

**Press** `[Printer]`

The Command window appears as follows:

```
Output to file ... ECHO.TEXT and Printer
Enter text or select function:
```

To send a text file from the Concept to the host computer,

**Press** `[SendFile]`

The Command window displays the following:

```
Enter file name:
```

Type the file name that you wish to send and press **[RETURN]**.

**Type** /VOL7/MYFILE.TEXT

**Press** **[RETURN]**

The following displays in the Command window:

Enter decimal ASCII character code to use with handshake.  
Example: enter "10" for a LineFeed, enter "0" for NO handshake: 10

The handshake that you choose depends on the system that is acting as the host computer. Enter the number for the ASCII character code to use, and press **[RETURN]**.

If for example you are tied onto a mainframe and you are using its editor to transfer your Concept file onto its system, and it prompts with ">" at the beginning of every line, you should look in the character set table in the chapter on character sets and find the ASCII number of the symbol. Since the number is 62, that is what you should enter by typing 62 and pressing **[RETURN]**.

The Command window displays the following:

Enter number of milliseconds delay desired after each line is sent: 40

Because some mainframes signal that they are ready to receive input even before they are actually ready problems arise. By setting a standard delay, problems associated with improper signals should disappear. If 40 milliseconds does not work properly you can change the delay by typing another value and,

**Press** `RETURN`

The Command window displays the following:

```
Sending file: /VOL7/TEXTGARB
is sent: 40
```

At this point the Concept is ready. Any input received by the Concept is sent to the display, and a file or printer as specified.

The `ClrWndow` function operates as it does in the Dispatcher or File Manager levels, and the `Exit` key stops the program and returns to the level that was current before the terminal program was run.

The terminal program does not automatically disconnect you from the other computer. You must log off of the other system to break communications.



## THE FLOPPY UTILITY PROGRAM

The program FUTIL in /CCUTIL is the floppy utility program. This program contains three utilities: the floppy formatter, the configure program, and the writeboot program. The floppy formatter allows you to set the density and number of sides, plus the sector size, the interleaving, and the skew. The configure utility allows you to set the logical interleaving, the skew, and the track offset, so that the floppy drive can read diskettes with non-Corvus formats.

Changing options in these programs can be dangerous if you do not know exactly what you are doing. The default settings are usually acceptable.

To enter the FUTIL program

**Type** /CCUTIL/FUTIL

**Press** RETURN

The following function keys appear while a corresponding menu appears in the System Window:

### FUTIL Program

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Formatr		Configur		Wr Boot					Exit

## The Floppy Formatter

The Corvus floppy diskette formatting utility is part of the FUTIL program. After entering the floppy utility program

Press **Formatr**

The Formatter program runs and the function keys appear as follows:

### The Floppy Formatter

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Format		WriteDir		Options					Exit

From this point, assuming that there is a diskette in the drive, the diskette can be formatted, the directory could be written on the diskette, or the formatting options could be changed.

**Formatter Options** The formatting options should not be used frequently, since they are already at the optimum settings for the Concept. To change any parameters

Press **Options**

The following menu and table appear on the screen:

Formatting options:		
Unit #	— Floppy unit number	9
Verify	— Verify format . . . . .	No
WriteDir	— Write Directory . . . . .	Yes
Density	— Density . . . . .	Double
SectorSz	— Sector size . . . . .	256
Sides	— Number of sides . . . . .	2
Interlv	— Interleave specification	2
Skew	— Skew factor . . . . .	6
Exit	— Exit to main menu	
	Tracks/side . . . . .	77
	Sectors/track . . . . .	26
	Block capacity . . . . .	2002
	Drive type . . . . .	8 inch

The function key labels change to reflect the menu in the System window.

The values in the option table are for the 8 inch floppy drive. If there is a 5¼ inch drive attached the Concept sees that drive and defaults to the following values: tracks/side - 80; sectors/track 16; block capacity 1280; and drive type 5¼ inch.

The options are as follows:

Unit # — the logical unit number of the diskette drive. This number defaults to 9 for the first drive, (and ten for the second).

- Verify — a sector by sector confirmation of the format of the diskette. After formatting each track the program reads each sector to check for errors. This option more than doubles the time required to format a diskette.
- WriteDir — tells the formatter whether or not to write the directory on the first six blocks of the diskette. The diskette must have a directory before files can be copied onto it.
- Density — the default is double density. Double density formats the diskette so that twice as much information can be stored in the same space. The old controller cards read only single sided, single density material.
- SectorSz — this is the number of bytes per sector. A double density diskette must have a sector size of 256 or 512, single sided diskettes can have 128, 256, or 512.
- Sides — the default is two, which requires both a double sided diskette, and a double sided diskette drive. A double sided diskette is capable of storing twice as much information as a single sided diskette.
- Interlv — along with skew determines the order physical sectors will be written on the diskette. The defaults of 2 for interleave and 6 for skew give the optimal access speed.

- Skew — see interleave above.
- Tracks/side — always 77 for the 8" floppies, and 80 for the 5¼ inch floppies. This number is displayed but cannot be changed.
- Sectors/track — this number is a function of the density and the sector size.
- Block capacity — this number is displayed and is determined by multiplying the number of tracks, the sectors per track, the number of sides, and the sectors per track, and dividing that number by the blocksize (512).
- Drive type — either 8 or 5¼ inch

Unless you have a specific configuration in mind you should not change any options other than the number of sides, the sector size, and the density. The other specifications should be all right for most purposes. The default settings are those of a double-sided double-density diskette.

Double-density settings must have a sector size of 256 or 512, while single density diskettes can be set at 128, 256, or 512. The maximum storage space available is double sided double-density with a sector size of 512. This results in approximately 1.6 megabytes for an 8 inch diskette, and 720 kbytes for the 5-1/4 inch diskette.

To change Sides from one to two

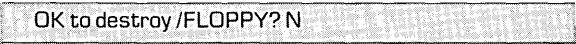
**Press**

The number changes in the option table. To change it back to two press **Sides** again. Verify, Write directory, Density, and Number of Sides all change back and forth automatically by pressing the key. Unit number, Sector size, Interleave, and Skew factor need to have a key pressed to enter a new value.

**Formatting a Floppy.** Formatting a floppy diskette is done from the Formatter level. After setting the proper options exit the option level. Be sure that the diskette to be formatted is in the diskette drive.

**Press** **Format**

The floppy diskette is automatically formatted. If the diskette has already been formatted with a valid directory one prompt must be answered. The following question appears in the Command window:



OK to destroy /FLOPPY? N

The default is No. If you do want to format /FLOPPY (or another name if the diskette is not named /FLOPPY)

**Press Y**

The floppy is then formatted. Displayed on the screen is a number that tells what track is presently being formatted. For the 8 inch drive the number of tracks is 77, and when the number reaches 76 the format is completed. For the 5¼ inch drive the number of tracks is 80, and when the number reaches 79 the format is completed.

If the Verify option was selected each track is read after being formatted and the System window appears as follows:

```
Formatting track 24
```

Then as follows:

```
Verifying track 24
```

The Formatting-Verifying messages switch back and forth with each track number from 0 to 76 (or 79) . If an error is found the speaker beeps and 'Read Error' appears above the Verifying track line.

In the Command window the following message appears:

```
C)ontinue R)etry Q)uit
```

You are prompted for the type of action you would like to take. Continue advances the formatter to the next track and aborts all attempts to format the current track. Retry attempts to format the current track again. If a read error appears a second time on the same track press C for Continue. Quit stops the formatting. If the diskette has too many bad tracks or if it must be perfect for your purposes Quit and try another diskette.

**Writing the Directory.** A floppy diskette that has already been formatted at some time can be initialized by writing the empty directory on it. The directory takes up the first six blocks of the volume, but must be there before the diskette can be used.

If the formatting program is used the directory is automatically written after the formatting process. The WriteDir option would have to be set to NO to stop this. When the directory is written the diskette is given the volume name /FLOPPY. That name can be changed using the RnamFile function in the File Manager.

A preformatted diskette needs to have the directory written. We suggest however that you format the diskettes with the Corvus formatting program, because our formatter is set to optimize the efficiency of the diskette drive. Diskettes that have been formatted using the formatter option of this program run faster than other diskettes. We suggest you purchase the less expensive unformatted diskettes.

To write the directory on a formatted diskette

**Press**

The diskette has the directory written on it in approximately one second. If there is information already on the diskette the following question appears in the Command window:

OK to destroy /FLOPPY? N



The default is No. If you want to write a new directory on the volume /FLOPPY

**Press Y**

The directory is written.

If you are using CP/M and want to create a CP/M volume floppy, you must format the floppy and then zero the directory according to the directions in "The Corvus Concept CP/M User Guide."

### **Configuring the Floppy Driver**

The Configure program is used to alter the diskette driver so that it reads diskettes with different formats. It is also used to tell the floppy driver what type of 5¼ inch floppy is in the drive.

The diskette driver does not have the capability of seeing when a 5¼ inch floppy drive has been changed. If the diskette is switched and it is not the same format including the sides, density or sector size, then the configure program needs to be run. This tells the driver what sort of diskette is being used. The default setting is double-sided double-density with a sector size of 256.

By changing the interleave, the skew and the track offset, you get a logical reordering of physical sector numbers, the sectors on the track, and the track numbers. This allows other types of diskettes to be read.

To enter the Configure program

Press **Configur**

The following menu displays in the System window with corresponding function keys:

Driver Configuration Options:	
Unit #	— Unit number of floppy . . . . . 9
Interlv	— Logical sector interleave . . . . . 1
Skey	— Logical skew . . . . . 0
TPI	— Tracks/inch . . . . . 96
Sec/Trk	— Sectors per track . . . . . 16
Trk/Side	— Tracks per side . . . . . 80
Sides	— Number of sides . . . . . Two
Exit	— Exit to main menu
StepRate	— Step rate . . . . . 4
Timeout	— Timeout counter . . . . . 65536
Reset	— Reset driver
Floppy characteristics:	
Sector size	. . . . . 256
Density	. . . . . Double
Drive type	. . . . . 5¼ inch

The floppy characteristics appear only if there is a floppy diskette in the diskette drive. Only the drive type shows up without a floppy. If the drive is an 8 inch drive the TPI option does not exist. The only other differences are that there are 26 sectors per track, and 77 tracks per side.

The unit number is important if there is more than one diskette drive attached to the Concept. To access the second diskette drive change the value of the unit number to 10.

The option settings can be changed using the corresponding function keys. We strongly recommend that you do not change anything other than the interleave, skew factor, and track offset. If you do not know precisely what you are doing your settings will probably not allow you to read any floppy diskettes.

Once the settings are changed and the driver is set other diskettes may be used. The driver does not change back to the standard Concept settings until the settings are changed using the program, or the Concept is rebooted. The settings must be changed back to those in the table above. This can be done using the Reset function. Whatever settings appear in the table, pressing the **Reset** key returns all the values to the default.

A large amount of UCSD p-System software is formatted with a 6-2-1 (interleave, skew, track offset) setting. To read these diskettes the settings must be changed from the Concept's 1-0-0. To do this

**Press** **Interlv**  
**Press** 6  
**Press** **RETURN**  
**Press** **Skew**  
**Press** 2  
**Press** **RETURN**  
**Press** **TrkOffst**  
**Press** 1  
**Press** **RETURN**

The driver values change, and the corresponding settings on the screen change too. Remember: these settings do not change until the system is rebooted, or they are changed back to 1-0-0 using the **[Reset]** key.

## Writing the Boot Code

Before you can boot from a diskette, you must make the diskette bootable. This can be done with the Writeboot utility in the FUTIL program.

To make a diskette bootable,

Press **[Wr Boot]**

The following displays in the System window:

```
Write boot options:
UnitNmbr — Floppy unit number: 9
FileName — File name . . . . . /CCUTIL/BOOT.FLOPPY
Write     — Write boot code
Exit     — Exit to main menu
```

This program uses another file that is found in /CCUTIL called BOOT.FLOPPY. number is not 9, press **[UnitNmbr]** and enter 10. If the default settings are correct,

Press **[Write]**

The System window displays the following message:

```
New boot written.
```

The boot blocks have been written on the diskette when the last line above displays. The diskette must also contain each of the following files from /CCSYS before it is bootable:

CC. KERNEL	DRV. DISPHZ	DRV. ADISK
CC. SETUP	DRV. DISPVT	DRV. FDISK
CC. DISPAT	DRV. KYBD	CSD. DEFAULT
CC. EXEC	DRV. SYSTRM	CSK. DEFAULT
CC. FILMGR	DRV. TIMER	ASSIGN
CC. WNDMGR		

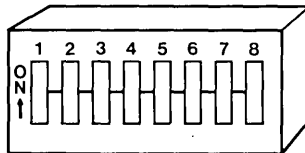
This chapter discusses peripheral connections for the Concept and how to change the orientation of the video display unit.

The first sections explain peripherals and boot types. The last section covers changing the video display unit orientation.

## TYPE OF BOOT

There are three different boot devices available on the Concept: diskette boot, local disk drive boot, and Omninet boot. The type of boot can be selected by setting the microswitches (boot switches) in the Concept drawer.

The boot switches inside the pull-out drawer on the back of the Concept must be set to the appropriate settings.



Switches 1 through 6 are used to set the Omninet address for each Workstation in a network. The Omninet address number displays next to the word "Station:" above the System window on the screen when the

network tap cable is connected. The proper settings for these switches can be found in the "Corvus Omninet Installation Guide."

Switches 7 and 8 are used to set the type of device from which the Concept will boot. The following table describes the available settings for these switches.

SWITCH 8	SWITCH 7	TYPE OF BOOT
On	On	Prompt for type of boot
Off	On	Boot from Omninet
On	Off	Boot from local disk

When the Concept is set to boot from Omninet, or a local disk, the Concept immediately presents the log-on prompt when it is powered on.

If you select the first type of boot mentioned in the table above, the following displays in the System window when you power-on the Concept.

```
CORVUS CONCEPT INITIALIZATION (#.#)
COPYRIGHT 1982 CORVUS SYSTEMS, INC.

ALL SYSTEM TESTS PASSED
SELECT BOOT DEVICE (D,F,L,O):
```

You are offered a choice of which type of boot you want. Press the key that corresponds to one of the letters described below to select the boot device.

D - allows access to the debugger, but only if you have the MACSbug debugger PROMs in the Concept.

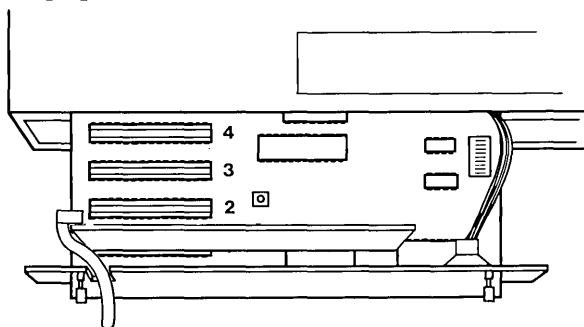
F - boots from the diskette drive.

L - boots from a local Corvus disk drive.

O - boots from an Omninet network.

## **DISKETTE CONNECTION**

The Corvus diskette drive interface card can be plugged into any of the four slots in the Concept pull-out drawer.



To boot from the diskette drive, set the boot switches to prompt for the type of boot, and press F with a bootable diskette in the drive.

## **LOCAL DISK DRIVE CONNECTION**

The Corvus disk drive can be plugged into any of the four slots in the Concept pull-out drawer. It is done in the same way as plugging in the card for the diskette drive.

To boot from a disk drive, set the boot switches as described in the first section of this chapter, or press L to the type of boot prompt.

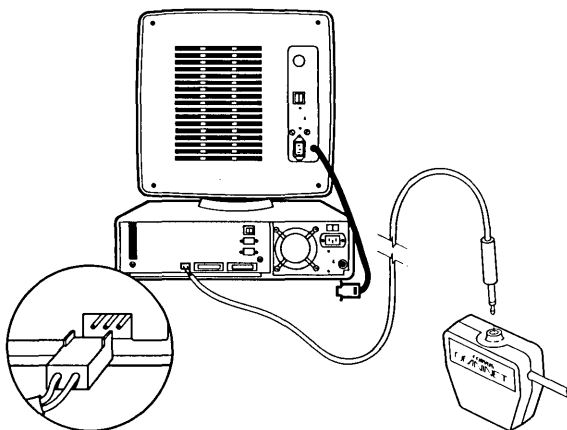


More than one disk drive can be connected to the Concept at the same time. If there are two or more disk interface cards connected to the Concept, it boots from the drive with an interface card in the lowest numbered slot.

For example, if you have a disk drive interface card in slot one and a second in slot two, the Concept boots from the disk drive with the interface card in slot one. Likewise, if you have a drive interface card in slot two and a drive interface card in slot four, the Concept boots from the drive with the interface card in slot two.

## OMNINET CONNECTION

The Omninet tap cable connects to the Omninet port on the back of the Concept base unit. The Omninet port is recognized as slot 5 by the Concept.



To boot from Omninet, set the boot switches as described in the first section of this chapter, or press O to the type of boot prompt.

## **RS-232C PORTS**

The Concept has two RS-232C ports on the back of the base unit. They can be used to attach printers, modems, etc., to the Concept.

### **Connecting a Local Printer**

This section details the equipment and procedures needed to connect a printer, which uses an RS-232C serial communications interface, to the Corvus Concept. It does not furnish the instructions for setting up any specific printer. It furnishes only those instructions needed to make a physical connection between the printer and the Concept.

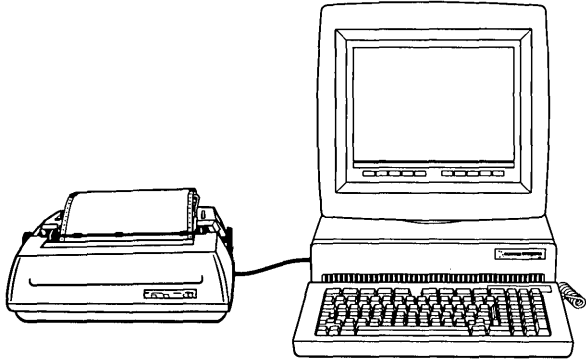
It is your responsibility to consult the reference manuals for your specific printer to find information such as switch settings, operational specifications, cautions and warnings.

You need the following equipment:

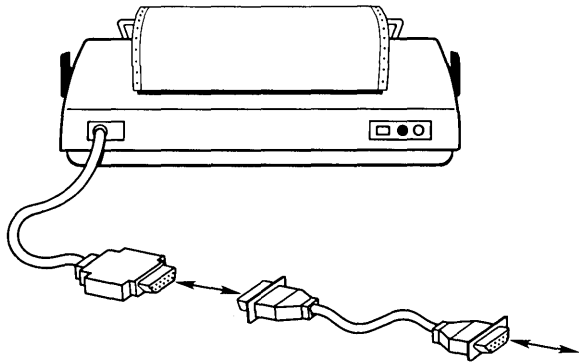
- A printer with an RS-232C serial asynchronous communications interface.
- A Modem Eliminator Cable, which is also called a Null Modem.

The Null Modem cable connects to the end of the RS-232C cable from the printer and then to the back of the Concept.

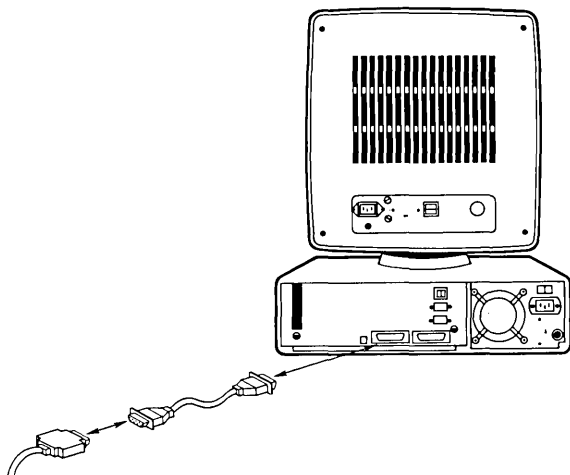
1. The printer should be close enough to the Concept so that the interface cable leading from the printer is slack.



2. Connect the Modem Eliminator Cable to the end of the printer interface cable.



3. Plug the opposite end of the Modem Eliminator cable into one of the RS-232C serial ports on the back of the Concept.



The serial port closest to the power switch on the Concept is port 1. The serial port furthest from the power switch is port 2.

Once you have connected the printer to the Concept, you should prepare it for use. This means that you should check the communication and operational settings of the printer. The items that should be examined are the following:

- Baud rate
- Parity
- Character transmission size
- Communication handshake protocol

Once you have the appropriate information, you should write it down. Next, consult the section titled "Setting Data Communication Parameters" in Chapter Seven of this guide, which explains how to set up a compatible communication link between the printer and the Concept.

If you are planning to use EdWord to print your documents, you must first set up your printer and Concept. Printing with EdWord is explained in "The Corvus Concept EdWord User Guide."

## **RS-232C INTERFACE SPECIFICATION**

Located in the rear of the Concept base unit are two RS-232C serial interface ports. These ports allow communication with various devices using the RS-232C interface standard. RS-232C is an interface standard developed by Bell Systems and standardized by the Electronic Industries Association. In the specification for this standard 25 lines are used to allow serial communication between Data Terminal Equipment (DTE) and Data Communication Equipment (DCE). DTE's are things such as Concepts and printers. DCE's are things such as modems.

Many devices, such as printers, come equipped with an RS-232C interface connection or have it as an option. It is by this interface connection that a physical link from a Concept to an RS-232C compatible device can be made, thus providing a basis for communications between the two.

This document details the interface specification as it is used with the Concept. Since various devices can be used with this interface standard, this document presents the information as if data is being sent from the Concept to a DCE, unless otherwise stated. Because printers are common devices that use this interface, applicable references to printers are included.

## **Overview**

Briefly stated, in order to communicate using RS-232C you need two items, 1) a physical connection, and 2) a protocol. The physical connection is provided most commonly by the subminiature D connector. This connector should contain the lines necessary for communication. However, more is needed than the physical connection between a Concept and a printer, and that's what a communications protocol is for. A communications protocol is a set of rules detailing the manner in which two devices communicate. A communications protocol consists of information such as how fast data should be transmitted, what signals are used to indicate when either the sender or receiver is ready, and whether or not there should be some sort of check to verify the data integrity. The various parts of a communications protocol are discussed later in this document.

This document first covers the hardware involved to make the physical connection, and then continues with a discussion of protocols.

## Line Assignments

Even though the specification for RS-232C details 25 lines, or pins, for data communication, most applications require only a few of these lines. The lines used by the Corvus Concept are shown in the following diagram. Also included in the diagram is the direction in which data is sent or received.

### Concept RS-232C Line Specification

( Direction:  $\leftarrow$  to Concept to DCE  $\rightarrow$  )

	1 *	-----	Earth Ground (NA)
	* 14		
	2 *	----->	Transmitted Data
	* 15		
	3 *	<-----	Receive Data
	* 16		
	4 *	----->	Request To Send
	* 17		
	5 *	<-----	Clear To Send
	* 18		
	6 *	<-----	Data Set Ready
	* 19		
Data	7 *	<----->	Signal Ground
Terminal Ready ->	* 20		
	8 *	<-----	Data Carrier Detect
	* 21		
	9 *		
	* 22		
	10*		
	* 23		
	11*		
	* 24		
	12*		
	* 25		
	13*		

A description of each of the lines used by the Concept follows:

**LINE 1:** Protective Ground (GND)  
**DIRECTION:** Not applicable

This line is used as an earth ground.

**LINE 2:** Transmitted Data (TxD)  
**DIRECTION:** To Data Communication  
Equipment

Data generated by the Concept is transmitted on this line to either a data communication device or a data terminal device.

**LINE 3:** Received Data (RxD)  
**DIRECTION:** From Data Communication  
Equipment

Data generated by the device is transmitted on this line to the Concept.

**LINE 4:** Request To Send (RTS)  
**DIRECTION:** To Data Communication  
Equipment

This line is used to indicate that the Concept is ready to transmit data.

**LINE 5:** Clear To Send (CTS)  
**DIRECTION:** From Data Communication  
Equipment

**LINE 6:** Data Set Ready (DSR)  
**DIRECTION:** From Data Communication  
Equipment



**LINE 7:** Signal Ground (SGND)  
**DIRECTION:** Not applicable

This line is a logical ground, also called a signal ground.

**LINE 8:** Data Carrier Detect (DCD)  
**DIRECTION:** From Data Communication Equipment

**LINE 20:** Data Terminal Ready (DTR)  
**DIRECTION:** From Data Communication Equipment

It is with these nine lines that the Concept is able to match most line configurations of devices using the RS-232C standard. However, even though there are nine lines, most applications require even fewer. The fundamental lines needed for communication are: 1) a line to transmit information from the Concept to the DCE, 2) a line for the Concept to receive information from the DCE, and 3) a signal ground. An earth ground is optional, though some equipment may require it.

These lines provide the bare essentials. What determines whether or not extra lines are needed is determined by how your communications protocol is set up. Communications protocol is considered next.

## **Protocols**

A communications protocol is a set of rules detailing the manner in which two devices communicate. In data communication there are different levels and types of protocols. This document covers what are considered low level protocols.

What constitutes the communications protocol for the Concept is a selection of baud rate, parity character size transmitted, and handshaking protocol. The term protocol can be applied to handshaking, since as stated previously, there are different levels of protocols. It just so happens that handshaking is a subset of the communications protocol.

When configuring a device to communicate with the Concept various parameters must be agreed upon between both devices. These parameters consist of information such as signals used to indicate when the sender or receiver is ready, how fast data should be transmitted, and whether there should be some sort of check to verify data integrity. Each of these parameters must be specified on both the printer and the Concept before printing can begin. If there is a mismatch between the two, data cannot be received without being garbled.

To set up the communications protocol for the Concept, you can use the SetDtaCm program. This command is accessed through the SysUtils level. This program is discussed in Chapter Seven.

## Handshaking Protocols

Handshaking protocols act as stop and go lights which coordinate the action between two devices. Handshaking provides a way to ensure that the sending device is not transmitting data while the receiving device is not ready to accept data. It also provides the means for a receiving device to indicate that it is ready for more data.

There are two ways to transmit a handshaking protocol. One way is to use a specific line of the RS-232C interface, apart from the lines through which data is sent. This is called a hardware protocol or line protocol. The other means is to use one of the transmitting or receiving lines to carry the protocol. This is called a software protocol.

Both means of sending a protocol are supported by the Concept. If you want to use a software protocol then you need only to connect the printer to the Concept, possibly using a Null Modem connector, and select a handshake protocol that is compatible with both devices. Software protocols are discussed under the heading "Software Handshaking Protocols."

If you want to use a hardware protocol you must be sure that the connector from your printer is appropriately configured for the selected line, and then you must choose the corresponding line protocol from the Line selection of the SetDtaCm program. Line protocols are discussed under the heading "Line Handshaking Protocols."

## **Baud Rate**

A baud rate is simply the speed in which data is sent from the Concept to the printer. The baud rate is how many elements, in this case bits, are transmitted each second. To simplify matters, assume each transmitted character is ten bits long. Thus we can relate to how fast characters are being transmitted. For example, 300 baud is 30 characters a second, and 1200 baud is 120 characters per second.

It is important to keep the baud rate in mind when setting up your printer and Concept to run with each other. For example if you set the Concept to send characters at 1200 baud, and you have the printer set to accept characters at 300 baud, you will either get an error from your printer, rebuking you for overwhelming it, or you will get nothing. In either case, it should be understood that the baud rate is important to match on both devices.

Selecting the appropriate baud rate for the Concept can be performed by using the SetDtaCm program, described in Chapter Seven.

## **Parity and Character Size**

Parity refers to a mechanism used to check the integrity of the data when it is sent, and character size refers to the number of bits sent that will be considered data.

Parity is a mechanism, when active, that uses one of the bits sent as data to make the binary sum of the data sent odd or even. Whether the sum is odd or even is determined by a setting on your printer, and can be matched by the Concept. For example, if even parity was selected by both devices, then the Concept would send data and with the parity bit make the sum even. The printer on the other hand would count the incoming data to see if the data is indeed even. If the sum of the data turns out odd then the printer indicates that an error has occurred, and will probably malfunction. If the sum is even then the communications continues.

Two other types of parity are: space no receive, and mark no receive. When space is sent a zero is used. If a one is received the printer or modem sends an error message, but the Concept does not receive it. The no receive is built in so that the Concept will not receive a retransmit order. Mark no receive is the opposite of space no receive. A mark is sent: a one is used, and a zero indicates an error.

Character size refers to the number of bits in the transmitted character.

## **Software Handshaking Protocols**

The Concept supports three main software protocols. The protocols supported are Xon/Xoff, Enq/Ack, and Etx/Ack. A brief and simplified description of how the protocols operate is as follows:

**Xon/Xoff:** Transmit on/Transmit off dc1 dc3

This handshake involves the Concept sending data to the printer until the printer tells it to stop sending by sending an Xoff. The Concept then waits until it receives an Xon character from the printer indicating that it is ready to receive more data.

**Etx/Ack:** End of text/Acknowledge

This handshake involves the Concept sending an Etx character after a block of data indicating the end of the transmission. When the printer reaches this character, it then sends an Ack character acknowledging that it is ready to receive more data.

**Enq/Ack:** Enquire/Acknowledge

This handshake involves the Concept sending an Enq character after a block of data indicating the end of the transmission. When the printer reaches this character, it then sends an Ack character acknowledging that it is ready to receive more data.

Remember: using a protocol requires that both devices agree on the handshaking used. It is evident that many problems would arise, for example, if the printer was waiting for an “end of text” while the Concept was waiting for a “transmit off.”

Consult your printer reference manuals to learn about what protocols you can select from, and how to select a desired protocol.

Selecting the appropriate handshaking protocol for the Concept is very simple, and is performed by using the SetDtaCm program discussed in Chapter Seven.

### **Line Handshaking Protocols**

Some printers may require that you use a line protocol, which means that handshaking is performed by use of a special line of the RS-232C interface. Unlike a software protocol, where the information about when to transmit and when not is on one of the data lines, a line protocol indicates a state change by simply changing the voltage of the selected line. The lines generally used for this purpose are lines 4 (Request To Send), 5 (Clear To Send), and 6 (Data Set Ready).

If you choose this mechanism, then you must select from the protocol options of the `SetDtaCm` label the appropriate line protocol. The line protocols available are Data Carrier Detect (DCD), Clear To Send (CTS) and Data Set Ready (Data Set Ready). The appropriate protocols associated with the line assignments are as follows:

Line 4 ----- Data Carrier Detect  
Line 5 ----- Clear To Send  
Line 6 ----- Data Set Ready

An additional factor of the line protocols is whether or not the signal on the line is normal or inverse.

Normal is a signal that is positive-true while inverse is negative-true.

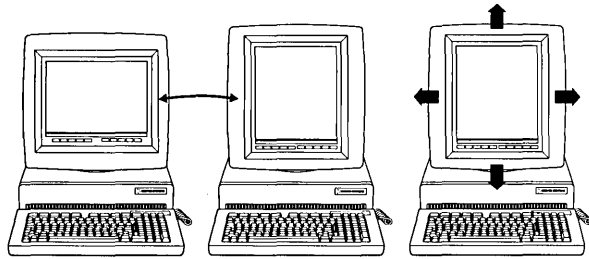
### **Null Modem Connector**

Nearly all printers using an RS-232C interface can be used with the Concept. However, certain printer connections with the correct communication protocol may not seem to work. Since you can select the correct communications protocol through the `SetDtaCm` program, you probably need either a Null modem cable or a Symmetrical cable.



## CHANGING SCREEN ORIENTATION

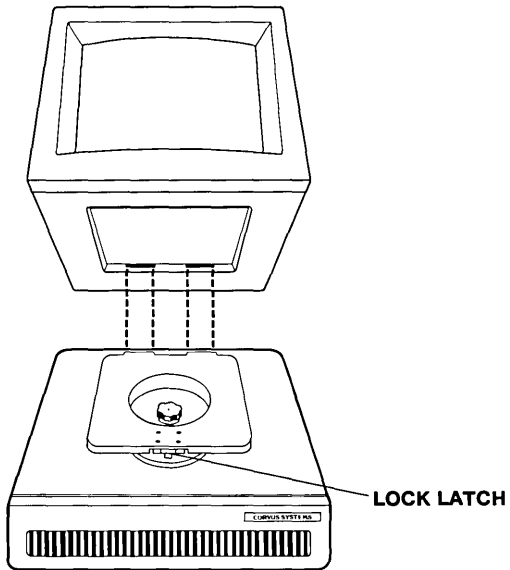
The Corvus Concept display unit can be arranged in different ways to suit your needs. Its orientation can be changed from vertical to horizontal, and vice versa, and in either position, you can tilt and swivel the display unit to your liking.



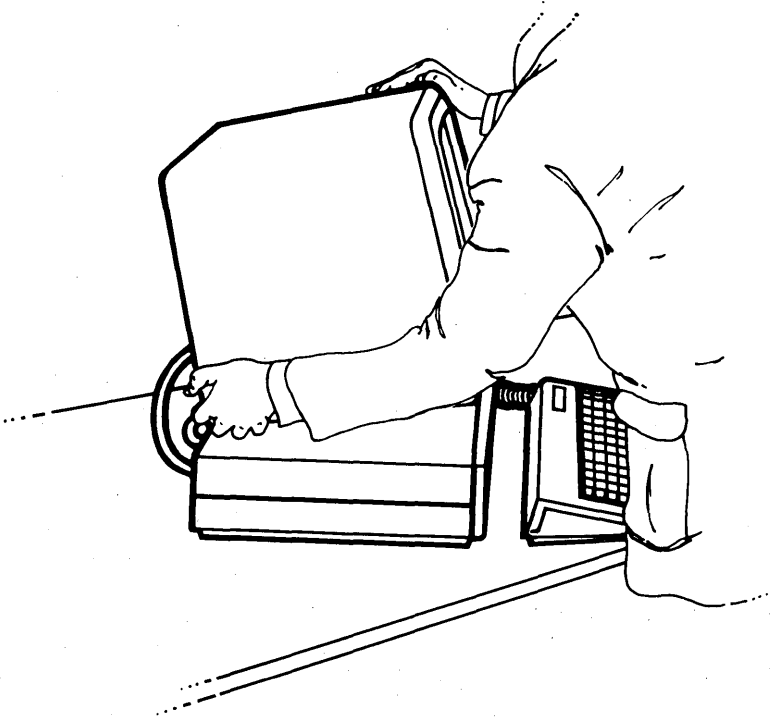
## Switching from Vertical to Horizontal

To change the Concept screen from vertical to horizontal, do the following:

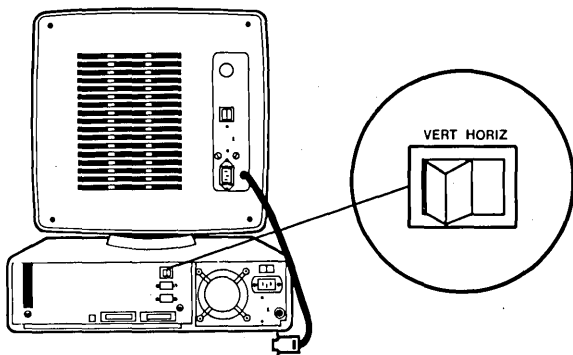
1. Stand in front of the Concept. Push the lock latch back, lift the front of the display slightly and slide it back about an inch.



2. Place your left hand on the back lower left corner. Place your right hand on the front upper right corner. Lift the display unit and turn it clockwise.



3. Lower the back end first and slide it forward to catch the two projections on the back of the display platform, then lower the front end. Pull the lock latch shut. Check to make sure the display unit is secure.
4. Press the right side of the screen orientation switch located above the video connector on the back of the base unit.

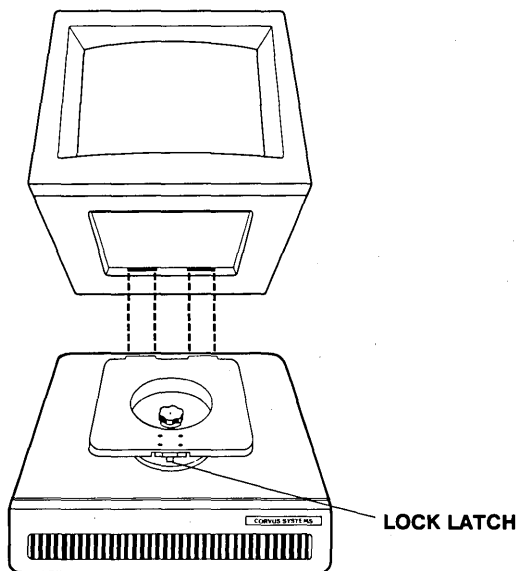


5. Hold down **CTRL** and Press **BREAK** to reboot.

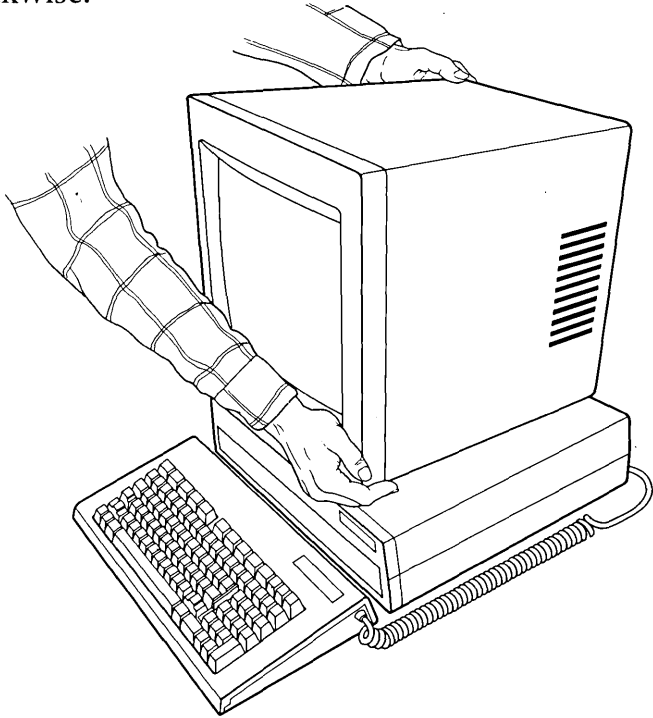
### Switching from Horizontal to Vertical

To change the Concept screen from horizontal to vertical, do the following:

1. Stand in front of the Concept. Push the lock latch back, lift the front of the display slightly and slide it back about an inch.

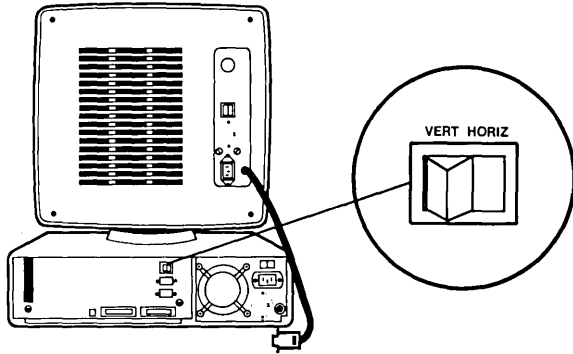


2. Place your left hand on the back upper left corner, and place your right hand on the front lower right corner. Lift the display unit and turn it counter clockwise.



3. Lower the back end first and slide it forward to catch the two projections on the back of the display platform, then lower the front end. Pull the lock latch shut. Check to make sure the display unit is secure.

4. Press the left side of the screen orientation switch located above the video connector on the back of the base unit.



5. Hold down **CTRL** and press **BREAK** to reboot.



# INDEX

## Label Index

AsgnDrvr .....	167, 168
ChngVol .....	49, 76
ClrWindow .....	29, 34, 49, 80, 82
ConcFile .....	49, 63, 64
Const II .....	29
CopyFile .....	50, 60-61
CreWindow .....	30, 50, 80, 85
Crunch .....	50, 77
DelWindow .....	80, 88
DletFile .....	50, 53-54
DletTemp .....	50, 56-57
DrvrVrsn .....	67, 169
EdChrSet .....	98
EdWord .....	30, 69-73
ExecFile .....	30, 150
Exit .....	35, 50, 80, 102, 115
FileMgr .....	30, 49-51, 118
Help .....	30, 34, 35, 50, 81
LdDispCh .....	70, 80, 81, 92
LdKybdCh .....	80, 81, 93
ListFile .....	50, 66
ListVol .....	30, 40, 50, 67
LogiCalc .....	30
MakeFile .....	50, 58-59
MountMgr .....	30, 51, 128
p-System .....	31, 157
Reboot .....	31, 35-36
Restart .....	31, 33
RevBkgnd .....	31, 51, 80-82, 114
RnamFile .....	51, 65
ScrlMode .....	80, 81, 83
SelWindow .....	31, 51, 80, 87, 88
SetDate .....	14, 15
SetDtaCm .....	159
SetTime .....	15, 16



SetVol .....	31, 39, 51
Spool .....	51, 118
SysUtils .....	13, 31, 98, 159
TestPtrn .....	80, 81, 84
TmpWindow .....	31, 51, 80, 81, 89
Userkeys .....	31, 153
WindowMgr .....	32, 80, 92

## General Index

Alternate Characters. . . . .	94-97
<i>see Character Sets</i>	
Assign Driver . . . . .	167-169
Asterisk . . . . .	53
<i>see Wild Cards</i>	
Boot Errors. . . . .	9
Booting. . . . .	9-10, 35-36
type. . . . .	189-191
Boot volume . . . . .	38
<i>see system volume</i>	
Change volume . . . . .	76
Changing Screen Orientation. . . . .	208-212
Character Sets . . . . .	91
Alternate sets. . . . .	94-97
alternate table . . . . .	96-97
<i>see Enhancements, Load Display Character Set,</i>	
<i>Load Keyboard Character Set</i>	
Clear window. . . . .	82
Cold boot. . . . .	36
<i>see booting</i>	
Command window. . . . .	12, 29
Compile. . . . .	74
Concatenating Files . . . . .	63-65
with wild cards. . . . .	63-64
Constellation II . . . . .	127
<i>see Mount Manager</i>	
Conventions. . . . .	3-5
Copying Files. . . . .	60-62
D Option . . . . .	62
Q Option . . . . .	62
S Option. . . . .	62
wild cards . . . . .	60-61
Create Window. . . . .	85-87
<i>see Temporary Window</i>	
Crunch. . . . .	50, 76-77
Current volume . . . . .	39
<i>see volumes</i>	
Cursor keys . . . . .	23
D Option . . . . .	62
Delete File . . . . .	50, 53-55
with wild cards. . . . .	53
Q option. . . . .	55
Delete Temporary File. . . . .	50, 56-57
Delete Window. . . . .	88

Despooling .....	124
<i>see Spooling</i>	
Diskettes, bootable	
<i>see Writing Boot Code</i>	
Connection .....	191
formatting	
<i>see Formatting diskettes</i>	
Dispatcher .....	12, 29
labels .....	11-12, 29-31
Display unit, changing screen orientation .....	208-212
characters on screen .....	28
configuration .....	24
screen interrupt .....	24
top of screen .....	11
Drive, local connections .....	191
Omninet connections .....	192
Driver, assigning .....	167-169
set printer .....	68
Driver Version, redirected .....	167-169
Editing Character Sets .....	98-116
entering the editor .....	98
changing parameters .....	99-102
edit mode .....	104-112
saving a file .....	112
test mode .....	113-115
exiting the editor .....	115
Editing a File .....	69-75
EdWord .....	30, 69-75
creating a workspace .....	70
exiting .....	75
labels .....	73
loading a file .....	73
in a startup file .....	151
re-entering the workspace .....	75
saving a file .....	74
Enhancements .....	69
loading the character set .....	69
stripping .....	62
Exclamation point .....	46
<i>see List Volumes</i>	
Executable Files .....	149-150
startup file .....	151-152
to load character sets .....	151-152
to load the printer driver .....	151-152
to enter EdWord .....	151-152

Executing a program	
<i>see Running a program</i>	
FFRMT	
<i>see Formatting Diskettes</i>	
File Manager.....	49
labels .....	49-51
Files, copying .....	60-63
creating .....	49, 57-59
editing .....	69-75
in other volumes .....	37, 52
names.....	52-53
listing.....	66
number per volume.....	51
recovering lost files .....	57-59
renaming.....	65
Formatting Diskettes.....	177-182
Function keys.....	20
definable .....	153-156
Function key labels.....	21
FUTIL	
<i>see Formatting Diskettes</i>	
Home volume.....	38
<i>see volumes</i>	
Interrupting screen display .....	24, 41, 67, 82
in p-System.....	157
Keyboard.....	19
<i>see Function keys</i>	
Label window.....	11-12, 28
List File .....	50, 66
redirecting.....	67
List Volumes.....	30, 40-48, 50
exclamation point listing .....	46-48
interrupting a listing .....	41
redirecting.....	67
slash listings .....	44-46
with wild cards.....	53
Load Display Character Set.....	69, 92
in a startup file .....	151-152
in a Userkey .....	153-155
set for EdWord enhancements.....	70
Load Keyboard Character Set.....	92
Logical units.....	38
Log-on .....	10

Modifier keys .....	23
Mount Manager .....	127-135
changing volume attributes .....	133-135
display volumes .....	130-131
display volumes mounted .....	131
exiting .....	135
help .....	128-129
labels .....	128
selecting a drive .....	132
Power-on .....	7-8
Print Window .....	89-90
Printing Files .....	68
from the p-System .....	128
physical connections .....	193-196
PRTWND .....	89-90
<i>see Print Window</i>	
p-System .....	31, 157-158
Q Option .....	55
<i>see Quiet Option</i>	
Question Mark .....	53
<i>see Wild Cards</i>	
<i>see Wild Cards</i>	
Quiet Option .....	55
with DletFile .....	55
Reclaiming volume space .....	76-77
<i>see Crunch</i>	
Redirecting Output .....	67
Rename File .....	51, 65
Restart .....	33
Reverse Background .....	82
RS-232 Ports .....	193-207
Running a program .....	32
S Option .....	62
<i>see Enhancements</i>	
Screen display	
<i>see Display unit</i>	
Scroll Mode .....	83
Select Window .....	87
Self-diagnostic .....	8
Set Data Com .....	158-167
set printer in startup file .....	151-152
SetDate .....	13-15
SetTime .....	15-17
Setting a volume .....	39
in a command line .....	151-152
Slash .....	44-48
<i>see List Volumes</i>	

Spooling .....	117
to a file .....	121-124
options .....	119-124
System Utilities, labels .....	13
System volume .....	39
<i>see Volumes</i>	
System window .....	11, 25-28
Temporary window .....	88
Terminal program .....	171-175
Test Patterns .....	84-85
UCSD p-System	
<i>see p-System</i>	
Userkeys .....	29, 153-156
Volume Utilities .....	135-148
backing up volumes .....	137-140
copying floppy diskettes .....	142-148
labels .....	136
restoring volumes .....	140-141
system status information .....	148
Volumes .....	37
number allowed .....	127
<i>see change volumes, Crunch, List volumes, Mount Manager,</i>	
<i>Setting a volume, Volume Utilities</i>	
VOLUTIL .....	135-148
<i>see volume utilities</i>	
Warm boot .....	36
<i>see booting</i>	
Warnings, power-on power-off .....	7
volume types .....	76
Wild Cards .....	53
with ConcFile .....	63
with CopyFile .....	60
with DletFile .....	55
with DletTemp .....	56
with ListFile .....	66
with ListVol .....	53
Window Manager .....	79
labels .....	80-81
Windows .....	11-12
clearing .....	82
creating .....	85-87
deleting .....	88
selecting .....	87
<i>see also System window, Command window, and Label window</i>	
Writing Boot Code .....	187
<i>see Formatting Diskettes</i>	

