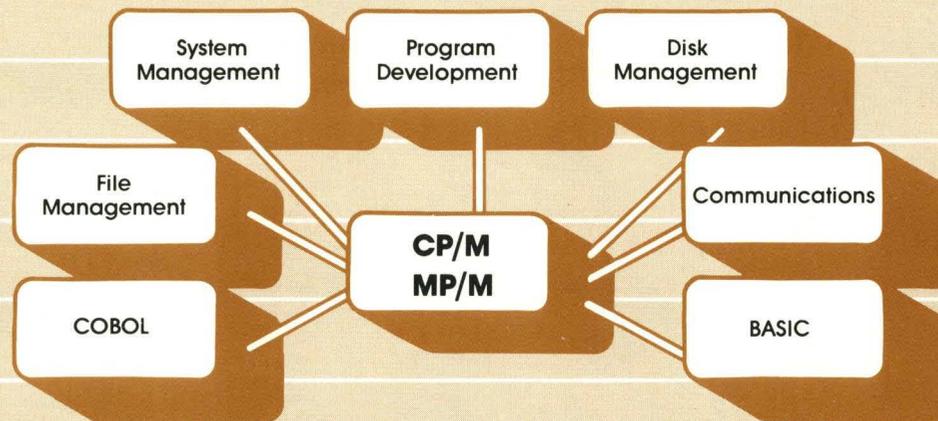


# CP/M<sup>®</sup>, MP/M<sup>™</sup> Extended Operations Manual





---

**CP/M<sup>®</sup> MP/M<sup>®</sup>, Extended  
Operations Manual**

---

September 1983



Revised September 1983  
Copyright ONYX Systems, Inc., 1983

All rights reserved. No part of this publication may  
be reproduced without prior written permission.  
Request additional copies from:

ONYX Systems, Inc.  
25 East Trimble Road  
San Jose, California  
95131  
(408) 946-6330

MP/M and CP/M are registered trademarks of Digital Research.  
Sundance and Onyx are trademarks of Onyx Systems, Inc.

Order No. D2005  
Printed in U.S.A. — 1M

---

### **NOTICE**

The information contained in this document is subject to change without notice.

Onyx Systems, Inc. MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Onyx Systems shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Onyx Systems assumes no responsibility for the use or reliability of software on equipment that is not furnished by Onyx Systems.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Onyx Systems, Inc.

---

## **PREFACE**

---

This manual deals specifically with the MP/M operating system, Version 2.1.2 and the CP/M operating system, Version 2.2.4.

You can use this manual to do the following activities:

- Learn how to operate the Onyx keyboard in the USING THE TERMINAL section.
- Become familiar with the basics of the MP/M or CP/M operating systems in the GETTING STARTED section. (A Glossary provides definitions of most commonly used terms.)
- Find detailed information regarding any of the MP/M, CP/M functions and utilities as supplied by Onyx in the USER REFERENCE section.
- Quickly look up system commands with the QUICK REFERENCE GUIDE.

Computer terms have been kept to a minimum to ensure an easy-to-read manual.

---

# TABLE OF CONTENTS

Section	Page	Section	Page	
<b>1. USING THE TERMINAL</b>		<b>2.7 BACKING UP FILES</b> .....	2-13	
Special Keys Used by MP/M, CP/M .....	1-2	Using a New Tape .....	2-13	
Special Keys Not Used by MP/M, CP/M .....	1-3	Using a Tape with Files on It .....	2-14	
Initializing the Terminal .....	1-3	Summary .....	2-14	
<b>2. GETTING STARTED</b>		<b>2.8 GLOSSARY</b> .....	2-15	
2.1 HOW TO USE THIS MANUAL .....	2-1	<b>3. USER REFERENCE MANUAL</b>		
2.2 STARTING THE SYSTEM .....	2-2	3.1 HOW TO USE THIS MANUAL .....	3-1	
Entering Commands .....	2-2	Organization .....	3-1	
Correcting Mistakes .....	2-3	Symbols .....	3-2	
2.3 FILE MANAGEMENT .....	2-4	Conventions .....	3-2	
Files .....	2-4	Terminology .....	3-2	
Disks .....	2-4	Additional Resources .....	3-2	
Device Names .....	2-5	3.2 USING THE SYSTEM .....	3-4	
Directories .....	2-5	File Names .....	3-4	
Special Characters .....	2-6	File Status .....	3-5	
2.4 RUNNING A PROGRAM .....	2-7	Disks .....	3-5	
Setting and Asking for the Date .....	2-7	Device Names .....	3-5	
Setting and Asking for the Time .....	2-7	Control Key Functions .....	3-5	
2.5 FILE OPERATIONS .....	2-8	3.3 COMMANDS AND UTILITIES .....	3-7	
Creating a File .....	2-8	ANSI	- Switch Sundance or Onyx Terminal from VT52 mode to ANSI Mode .....	3-8
Displaying a File .....	2-9	ARCHIVE	- Logical File Backup .....	3-9
Renaming a File .....	2-9	ASM	- 8080 Assembler .....	3-16
Copying a File .....	2-9	ASSIGN	- Set/Display Peripheral Assignments .....	3-17
Printing a File .....	2-10	CCREADY	- Control-C Ready, Used to Enable Type Ahead .....	3-19
Erasing a File .....	2-10	COL80/ COL132	- Switch Sundance/132 and Onyx DT/132 Terminal between 80 and 132 Column Mode .....	3-20
Summary .....	2-10			
2.6 WORKING ON ANOTHER DISK .....	2-11			
Running a Program .....	2-11			
Copying Files .....	2-11			

Section		Page	Section		Page
DATE	- Set/Display System Date . . .	3-21	PROTOCOL	- Set/Display Serial Port Protocol . . . . .	3-59
DBACK	- Logical Disk Backup to Tape . . . . .	3-22	PUB/PRIV	- Enable/Disable Public Files in User 0 . . . . .	3-61
DDT	- Dynamic Debugging Tool . . . . .	3-24	QERA	- Query User to Erase File(s) . .	3-62
DDUMP	- Dump Physical Disk Sectors . . . . .	3-25	REN	- Rename File . . . . .	3-63
DIR	- Display Directory Contents . . . . .	3-27	SAVE	- Save Program Memory Image in Disk File . . . . .	3-64
DLOAD	- Logical Disk Load from Tape . . . . .	3-28	SPARE	- Update/Display Spare Sector Directory . . . . .	3-65
DO	- Multiple Commands per Line . . . . .	3-30	SPEED	- Set/Display Baud Rate for Serial Devices and Tick-Rate for Real-Time Clock . . . . .	3-67
DSTAT	- Disk Usage Display . . . . .	3-31	STAT	- Set/Display Disk and File Status . . . . .	3-69
DUMP	- File Dump in Hexadecimal Format . . . . .	3-32	SUBMIT	- Command File Processor . .	3-73
DVFY	- Verify Disk Backup Tape . . . .	3-33	TIME	- Set/Display System Time . . .	3-75
ED	- Create/Edit ASCII File . . . . .	3-35	TX/RX	- Transmit/Receive Files to/from Another Computer . . . . .	3-77
ERA	- Erase File . . . . .	3-37	TYPE	- Display ASCII Text File on Console . . . . .	3-83
ERASEALL	- Erase Logical Disk . . . . .	3-38	UNERA	- Recover Erased File . . . . .	3-84
FCOMP	- File Comparison . . . . .	3-39	USER	- Set Current User Directory Number . . . . .	3-85
FDUMP	- File Dump in Hex and ASCII Formats . . . . .	3-40	VIS	- Remove SYS and R/O Status from File(s) . . . . .	3-86
FILESET	- Create Alphabetical List of Filenames for use with ARCHIVE . . . . .	3-41	VTSETUP	- Configure CP/M to Switch Sundance or Onyx Terminal to 80 Column ANSI Mode on Warm Boot . . . . .	3-87
FINDALL	- Search Disk for File(s) . . . . .	3-43	VT52	- Switch Sundance or Onyx Terminal from ANSI Mode to VT52 Mode . . . . .	
FREEZE	- Save System Configuration . . . . .	3-44	XSUB	- Extend Command File Processor . . . . .	3-87
INVIS	- Move Files to User 0 and Set Status to SYS and R/O . . . . .	3-45	3.4 HARDWARE ERROR MESSAGES		3-90
LOAD	- Create Executable Program from Assembler Output . . . .	3-46	Disk Error Messages		3-90
LOADREL	- Load Software Release Tape from Onyx . . . . .	3-47	Tape Error Messages		3-90
MOVEF	- Move File(s) Between Users .	3-49			
OPTIONS	- Set System Options . . . . .	3-51			
PATCH	- Patch CP/M . . . . .	3-53			
PIP	- Copy, Print and Combine Files . . . . .	3-54			

Section	Page
3.5 OTHER SDT FUNCTIONS .....	3-92
Using the SDT .....	3-92
3.6 STANDARD ASCII CHARACTER SET .....	3-94
3.7 SUBJECT INDEX .....	3-98

#### **4. QUICK REFERENCE GUIDE**

### **TABLES**

Table	Page
1-1 Mode B Parameter Defaults for the Sundance/132 and the DT/132 .....	1-3
1-2 Mode C Parameter Defaults for the Sundance/132 and the DT/132 .....	1-4
1-3 Parameter Defaults for the Sundance 80 and the DT/80 .....	1-4
3-1 User Reference Manual Terminology .....	3-3
3-2 ARCHIVE Commands .....	3-11
3-3 Physical/Logical Track Assignments .....	3-25
3-4 PATCH Dialogue Example .....	3-53
3-5 Abbreviated PIP Commands .....	3-55
3-6 PIP Parameters .....	3-56
3-7 TX Error Messages .....	3-79
3-8 Disk Error Messages .....	3-90
3-9 Tape Error Messages .....	3-91
3-10 Standard 7-bit ASCII Character Set .....	3-94

### **ILLUSTRATIONS**

Figure	Page
1-1 Key Locations .....	1-1





---

Section 1

# **Using the Terminal**

---



# 1. USING THE TERMINAL

---

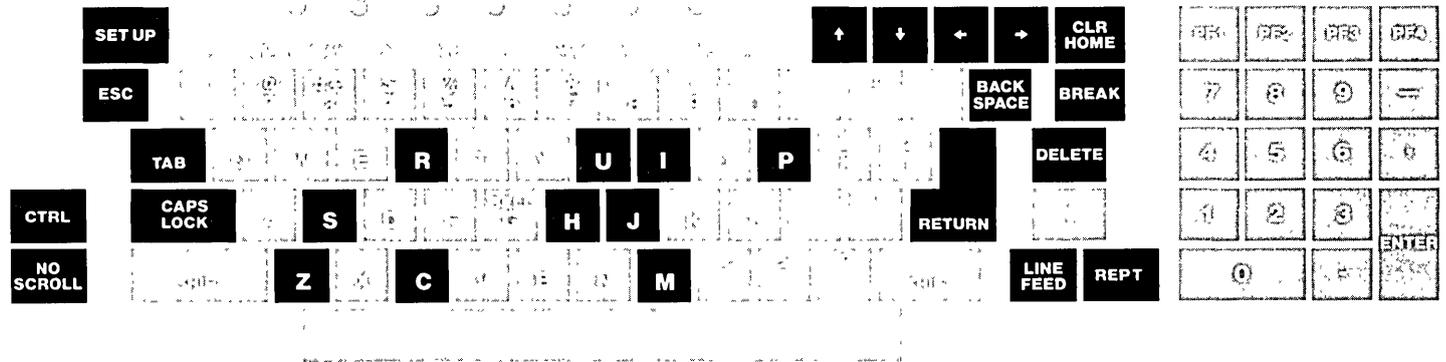


Figure 1-1 Special and Control Key Function Keys

---

Before trying to use the system let's take a minute to look at the Onyx terminal keyboard and discuss some keys you probably haven't seen before. If you are using a different terminal you may have these same keys and they will probably perform in the same way. You should test them out and note any differences. You will find that most of the operations performed by special keys can be done in more than one way, so if a key is not part of your keyboard, you will still be able to perform the operation.

For the most part the Onyx keyboard looks like a typewriter, but on the left side there are six keys labeled

'SET-UP',  
'ESC', 'TAB',  
'CTRL', 'CAPS LOCK',  
'NO SCROLL'

that are new.

On the right side there are the 'ARROW' keys, each labeled with an arrow pointing in a different direction, and the keys

'CLR HOME',  
'BACK SPACE',  
'BREAK', 'DELETE',  
'RETURN', 'LINE FEED',  
'REPT'

There is also a separate set of keys clustered together on the right side that form the numeric keypad. We don't need them to use the system; in general a key in the numeric keypad is identical to its counterpart in the main set of keys. For example, pressing '9' will enter a '9' in either case. All of these special keys are described here including those not used by the system. Take a moment to locate these keys on your keyboard. You can use figure 1-1 to help you.

## Special Keys Used by MP/M, CP/M

### ESC

When the system is waiting for a command, pressing it causes a '^' to be displayed. You can erase this by using the 'BACKSPACE' key. When the system is in the middle of an operation, ESC stops the operation.

### TAB

The TAB key works like the TAB key on a typewriter. It causes the cursor to move to the right to the next tab stop. The tab stops are set at every 8th column (8, 16, etc.). The operating system translates TABs into spaces and ignores them.

### CTRL

A most important key, used in the same manner as the SHIFT key. When you want to type a capital letter, you would press the SHIFT key plus any letter in the alphabet. Like the SHIFT key, you use the CTRL key together with another key. The combination of pressing the CTRL key with another key at the same time is called a CTRL function. These are shown throughout as CTRL + \_\_\_ where \_\_\_ is a letter of the alphabet. There are a number of them. Each is a command. These functions are all described in the User Reference Manual (Section 3.2).

### CAPS LOCK

CAPS LOCK is like the SHIFT LOCK on a typewriter. It causes all the alphabetic keys to be upper case. The other keys are not affected. CAPS LOCK stays locked down when you press it. It returns to the normal position by pressing it again. The operating system treats all alphabetic characters as if the CAPS LOCK key were pressed down, interpreting all letters as upper case.

### NO SCROLL

On the Onyx DT/132 and Sundance/132 terminals, NO SCROLL stops the output being displayed on the screen. This is useful when you are displaying a long list that is larger than what can be shown on the screen. Pressing the NO SCROLL stops the operation; you can restart it by pressing it again. When the system is not in the middle of an operation, pressing it causes a '^S' to appear on the screen and the ONLINE indicator to blink. You can type on the keyboard but nothing will happen. Press it again and a '^Q' appears. The ONLINE light will stop blinking. Erase these characters with the BACKSPACE key.

### BACKSPACE

Used to erase the character you've just typed. When you press it, the 'cursor' moves one character to the left and the character disappears from the screen. This is the easiest way to correct typing mistakes.

### DELETE

Used to remove or to erase the character you've just typed. Instead of it disappearing, the system displays it again. So if you type:

abcd and press the 'DELETE' key to erase the 'd'

the screen will show

abcdd

but as far as the system is concerned, only the

abc

has been typed.

### RETURN

A very important key. Any command not issued using the CTRL key ends with the RETURN key. It is functionally identical with the ENTER key in the numeric keypad. RETURN signals the system that the command is complete. The system responds to the RETURN either with the operating system prompt (>) showing that the command has been completed or by asking for more information. When you press the RETURN key while running a system utility or an application program, the cursor moves to the beginning of the line.

## LINEFEED

Used just like the RETURN key. When used by a system utility or application program, the LINEFEED key causes the cursor to move down one line.

## Special Keys Not Used by MP/M, CP/M

The following keys are not used at this time, but are described here because they may have some effect if pressed accidentally.

## SET-UP

If you press it accidentally the keyboard/display stops communicating with the computer. Press it again to restore normal operation. A '^S ^S ^Q' will appear. Use the BACKSPACE key to erase these characters.

## ARROW KEYS

If you press one of these keys, an '^([A', '^([B', '^([C', or '^([D' appears. They can be erased with the BACKSPACE key. Some programs use these keys to move the cursor in the direction of the arrow—i.e., left/right, up/down.

## CLR HOME

It causes the cursor to move to the upper left hand corner of the screen, regardless of what is displayed there. When you start typing characters, they will replace any characters that were there. Pressing RETURN will cause the operating system prompt (>) to appear.

## BREAK

Pressing it will cause a '^@' to appear on the screen. It can be erased with the BACKSPACE key.

## REPT

Nothing happens when you press this key.

## Initializing the Terminal

If you are using a Sundance or Onyx Terminal on your system, its operational parameters have been initialized to run with CP/M or MP/M. You do not have to make any changes.

Some parameter values are optional. You may select either value depending on your preferred mode of

operation. Consult the Sundance Hardware Operations Manual (Terminal Reference Section) for explanations on all terminal parameters and how to change them.

The default values are shown in Tables 1-1 and 1-2 for the Sundance/132 and DT/132. Table 1-3 shows the parameters for the Sundance/80 and DT/80. Parameters you may change without affecting CP/M and MP/M have 'optional' shown after them.

**Table 1-1**  
**Mode B Parameter Defaults for**  
**the Sundance/132 and the DT/132**

Parameter	Default Value
Scroll Style	Jump
Auto Repeat	On (optional)
Screen Background	Dark (optional)
Cursor	Block (optional)
Margin Bell	Off (optional)
Keyclicks	Silent (optional)
ANSI/VT52	ANSI
Auto XON-XOFF	On
Shift and 3	# (optional)
Shift Wraparound	On
Newline	Off
Interlace	Off
Parity	Odd
Parity	Disable
Bits per character	8
Screen Frequency	60/50 Hz as needed.
Flags	0 (optional)

Use the "7" key to set the transmit speed and the "8" key to set the receive speed. Either key causes the speed to advance to the next highest value. (These speeds may be set at anytime during Mode B.) The screen displays the current value of both parameters at the bottom of the screen (T SPEED for transmit and R SPEED for receive). Both should be set at 9600 baud.

**Table 1-2**  
**Mode C Parameter Defaults for**  
**the Sundance/132 and the DT/132**

<b>Parameter</b>	<b>Default Value</b>
Smooth Scroll Speed	Normal
Scroll Key Style	VT100
Aux Numeric Keypad	Normal
Form Feed	Line Feed
Control Characters	Executed
Incoming XOFF	Disable
Cursor	Visible (optional)
Screen Data on 80-132 Changes	Preserve (optional)
RH of DW lines	Preserve (optional)
Erase Page Extent	Within Margins (optional)
Tab Motion	Direct (optional)
CRT Saver	On (optional)
Half/Full Duplex	Full Duplex
RTS-CTS Handshake	Disable
8th Bit on Transmit	Space
Stop Bits on Transmit	1

**Table 1-3**  
**Parameter Defaults for**  
**The Sundance/80 and the DT/80**

<b>Parameter</b>	<b>Default Value</b>
Main Port Speed	9600
Aux Port Speed	Not supported
Main/Aux Parity	Odd
RCVR Parity Check	Off
Video Attributes	Off (optional)
Respond to XON/XOFF	No
Generate XON/XOFF	No
Duplex	Full
KYBD Auto Repeat	On (optional)
Keyclick	Off (optional)
Shift and 3	# (optional)
ANSI Cursor Key	Off
Cursor	Block (optional)
Cursor	Blinking (optional)
Scroll	Jump
Background	Dark (optional)
Newline	Off
Autowrap	On
Mode	ANSI
Line Freq	60/50 Hz as needed



---

Section 2

**Getting Started**

---



## 2.1 HOW TO USE THIS MANUAL

---

This Getting Started Manual is designed to give you the basics so you can start using your system right away. Throughout the manual there will be examples of what the operating system displays on the screen and what you should type on the keyboard. To help you, we show in **bold** the information you are asked to type. Type only what is in **bold**, and press only the keys you are asked to press.

This section provides information for starting the system under both MP/M and CP/M as supplied by Onyx. Most of the information in this section applies to both operating systems. However, some information applies only to CP/M. Whenever a comment in a description applies only to CP/M, the text is distinguished by a gray half-tone background.

IMPORTANT: When you see

CTRL + C

it means hold down the CTRL key while typing a 'C'. Similarly with other letters (I, X, Z, etc.).

When you see

and press RETURN

it means press the RETURN key.

In the text we will always refer to the keys by name — the label on the key itself. We surround the alphabet and number key names with single quote marks: '. Don't type the '; they are used only to identify the key.

Also, throughout the exercises in this manual you will be asked to create, copy and transfer files on your disks. (Do not worry right now if you do not know what this means.) However, you do need to understand that when creating, copying, or transferring files we will ask you to add your initials to the file name. Then other individuals in your company can use this Getting Started Manual. Your initials will make the file names unique while you perform the exercise.

Before you do the exercises in this manual, you may want to read this document completely through so you have an idea of what you will be doing in the next hour.

Throughout this manual and the other system documentation you have received with your computer, you will find many new terms. To help you start using them we have included a Glossary at the end of this manual (Section 2.8). Consult it whenever you encounter a term that is new to you. These terms are used extensively by people who work with computers. You'll be using them too because they describe operations and concepts that are basic to working with a computer.

## 2.2 STARTING THE SYSTEM

---

The following steps will get the operating system running.

### 1. Turn on the system.

If you have a Sundance, the switch is on the back near the top of the unit. If you have a C5001 or C8001 system, turn on the terminal, and with the key, turn the key switch on the front panel to the right. If you have a System 2000, the switch is on the back panel in the lower-left corner.

If nothing happens, make sure the unit is plugged into a live outlet.

### 2. Press the reset button.

If you have a Sundance, the reset button is under the protruding panel on the back panel. (You can't see it, you can only feel it with your fingers.) If you have a C5001 or C8001, the reset button is on the back panel in the top-right corner. If you have a System 2000, the reset button is the red button on the back panel.

The screen displays an ONYX self-test message similar to the one below:

```
ONYX SU self test V1.2
```

### 3. When the self-test is complete, an "\*" will appear. Press the RETURN key. This will load the operating system.

The operating system automatically processes the command SUBMIT STARTUP. (See Section 3.3 for a description of the command SUBMIT and the possible uses of the STARTUP.SUB file.)

4. A series of messages appears on the screen. When the operation finishes, the operating system prompt appears. The '>' signals that the system is waiting for a command. The letter to the left of the prompt identifies the disk in use; you will learn more about that a little later. The number to the left of the letter is the MP/M user number. See the MP/M User Guide for a description of user numbers.

```
0A> _____ System Prompt  
      | _____ Disk Name  
      | _____ User Number
```

On CP/M systems, the operating system prompt consists of '>' and the disk name.

```
A> _____ System Prompt  
      | _____ Disk Name
```

### Entering Commands

Each command to the operating system consists of:

- the command itself
- the data the command is to process
- the RETURN which tells the system to perform the command

A space separates the command from the data. The data for each command is in a certain order and format; we'll describe the formats when we describe the commands.

## Correcting Mistakes

If you make a typing mistake, use the BACKSPACE key to erase the last character you typed. The system will remove the character from the screen.

You can erase an entire line that way or you can use CTRL +U to erase the entire line. CP/M will move the cursor down one line so you can start over. It won't erase the line from the screen or display the '>' again.

Try typing 'bbbb' on the line after the prompt A>, then press CTRL +U (hold the CTRL key down while pressing the 'U' key), then press RETURN. Notice what happened.

When you enter a command that CP/M understands, it will process it and respond with the '>'. If there is an error, CP/M will display the command as you typed it followed by a question mark (?) and then the '>'. Check the command carefully; chances are you misspelled the command, or left out something, or put the data in the wrong order. Some errors cause CP/M to display an error message. This message should be used to determine the nature of the error you made.

CP/M translates all lower case alphabetic characters to upper case when it processes a command and its data. You can leave the CAPS LOCK key down so that what is shown on the screen appears exactly as CP/M uses it, but it's not necessary.

## 2.3 FILE MANAGEMENT

---

### Files

CP/M is a file management system. That means that all the data you want to store or manipulate or display is stored in some type of file. System utility programs and application programs manage the data; CP/M manages the files. So most of the commands to CP/M are for handling files. You can display them, print them, erase them, copy them, etc.

Each file has a name that you give it when the file is created. Files are created either by system utility programs, like PIP or ED, or by application programs. The file name is up to you but there are the following limitations:

1. The name can be from 1 – 8 characters, alphabetic, numeric, or \$. Names can be longer but only the first 8 characters are used and displayed.
2. File names should be unique but we'll discuss later on how you can have two files with the same name.
3. **No periods (.), commas (,), asterisks (\*), question marks (?) or spaces** can be part of the file name as these characters have special meanings.

A file may have a secondary name, sometimes called an extension, as well as the primary name we've just described. The secondary name follows the primary name and is separated from the primary name by a period (.). It has the same restrictions as the primary name but it is only three characters long.

The secondary name is used to help you remember the type of file. For example, a secondary name TXT can mean that the file contains a document or a description. A data file to be used by an application program could have a secondary name DAT. Two files may have the same primary name as long as they have different secondary names; two files may have the same secondary name as long as the primary names are different.

Some possibilities are:

XYZ.	no secondary name
XYZ.TXT	} two files with same primary name
XYZ.DAT	
ABC.DAT	} two files with same secondary name

Some commonly used file types are:

.BAK	a backup file
.DOC	a document file
.DAT	a data file
.BAS	a file containing a BASIC program
.COM	a file containing a program ready to run
.\$\$\$	a temporary file

The User Reference Manual and the Quick Reference Guide list the most common ones. You can create your own. Just as with commands, CP/M makes no distinction between upper and lower case alphabetic characters; 'd' and 'D' are equivalent.

### Disks

In order to store files on a physical disk CP/M organizes it as a set of logical disks. These are like the individual file cabinets you have in your office. There can be as many as sixteen (16) logical disks on the system. Each has a name — A, B, C, D,....etc. CP/M always lets you know which disk it is using; the A> that you saw after you started up the system means that CP/M is using disk A.

TYPE:

B: and press RETURN

Notice that now the prompt is B>. This means that CP/M has switched to disk B. To get back to disk A:

TYPE:

A: and press RETURN.

What happened? You are now using disk A.

## Device Names

CP/M treats all devices as if they were files. Devices are other pieces of equipment associated with your computer. That means it gives each a name, the physical device name. You have already used the disk names. Some of the common device names are

TTY: keyboard  
CRT: display  
LST: printer

You can specify a device name anytime instead of a file name. We'll use device names later on.

## Directories

In order for CP/M to keep track of and locate the files on a disk, it keeps lists of file names and their locations. These lists are called directories. You can think of them as the drawers in a file cabinet. There are sixteen (16) directories for each logical disk, named USER0, USER1, USER2,....,USER15. When a file is created its name and where it is located are stored in one of the directories. So in order to locate a file you need to know the directory it's listed in and the disk it's stored on.

When you started up the system the process ended with the message:

```
Current user = 0  
A>
```

This means you are working on disk A and in the directory USER0. Let's see the list of files in directory USER0.

TYPE:

DIR and press RETURN

CP/M will display the names of all the files listed in the USER0 directory. Normally all the system software files are stored on disk A in the USER0 directory. That's because USER0 is special; files stored there can be used regardless of the directory you're working in. We'll talk more about that later.

Let's change directories.

TYPE:

USER 1 and press RETURN

Look at the files listed in the USER1 directory.

TYPE:

DIR and press RETURN

Notice that the files listed in USER1 are the same ones that were listed in USER0 (not necessarily in the same order).

TYPE:

B: and press RETURN

TYPE:

DIR and press RETURN

What happened?

The files listed in USER1 on disk B are not the same as those that were displayed when you were on disk A. That's because the directories only list the files stored on a particular disk. This is one way CP/M protects files by having more than one logical disk and different directories on each one. We can use the files in USER0 on disk A but we'll do that later.

Go back to disk A. Do you remember how?

TYPE:

A: and press RETURN

## Special Characters

Many times we want to work with a group of files, for example, all the files that are associated with a particular application. CP/M uses two characters, '?' and '\*' to help you do that. Let's list all the utility program files. They all have a secondary name COM.

TYPE:

**DIR \*.COM** and press **RETURN**

CP/M will list all the files with a secondary name of COM. The '\*' character acts as a wild card (poker anyone?). In this instance it stands for the 1 to 8 characters that may make up the primary file name.

Let's use the '\*' to list all the file names that begin with 'A'.

TYPE:

**DIR A\*.\*** and press **RETURN**

What is the response?

Notice that while all the file names begin with 'A' they have different lengths. Again, the '\*' stands for a variable number of characters, this time in both the primary and secondary names.

Now let's try the '?'.

TYPE:

**DIR ?X.COM** and press **RETURN**

The '?' substitutes for a single character in a file name. You can use both the '?' and the '\*' at the same time.

TYPE:

**DIR ??.\*** and press **RETURN**

This combination displays all the files with a two character primary name and any secondary name.

## 2.4 RUNNING A PROGRAM

---

To run a program, either a utility program or an application program, all we have to do is tell CP/M its file name. You saw the names of all the system utility programs when you used the DIR command. Each one has a secondary name COM. CP/M expects that a program that is ready to run will have a COM secondary name so that all you have to type is the primary file name.

Now let's run some of these system utilities.

### Setting And Asking For The Date

Having the system maintain the calendar date for you is very useful, especially if you want to have it printed on your reports. The format of the date is MM/DD/YY. MM is the month, DD is the day, and YY is the year. You don't need to type a leading 0 if the month or day is less than 10. 9/7/81 is perfectly OK. Now let's set the date.

TYPE:

**DATE MM/DD/YY** and press **RETURN**

The same utility program will display the date for you. Run the 'DATE' program again.

TYPE:

**DATE** and press **RETURN**

CP/M responds with the date you have just set.

### Setting And Asking For The Time

The time is stored as HH MM SS. HH is the hour (24 hour clock), MM is the minute, and SS is the second (optional). Like the date you need not type leading 0s. You must leave a space between the HH, MM, and SS. Now run the utility program that sets the time.

TYPE:

**TIME HH MM SS** and press **RETURN**

To have the time displayed, run the TIME utility program again.

TYPE:

**TIME** and press **RETURN**

CP/M responds with the current time.

## 2.5 FILE OPERATIONS

---

### Creating A File

We have learned how to list the file names; let's try working with an individual file. To do that we're going to create a little test file just to play with.

Files are usually created using the editor (ED) for program source files, the word processor (WS) for document files (or program source files), or an application program for data files.

The PIP utility program can also be used to create a file. PIP stands for Peripheral Interchange Program; it is the utility program for copying files on the disk. Let's run the PIP utility.

TYPE:

**PIP** and press **RETURN**

PIP has its own prompt, the '\*'. The '\*' means that PIP is waiting for a command. PIP commands look like this:

**TO=FROM(parameters)**

TO is the name of the file the data is to be stored in, FROM is the name of the file the data is stored in now, and the parameters describe how the transfer is to take place. Parameters are optional; we won't use them until later.

We will use only the simplest form of PIP commands; they are described in detail in the User Reference Manual and the Quick Reference Guide.

The command we are going to use tells PIP to create a file called 'TEST'.\_\_\_\_'

↑  
your initials

and store in it data from the file 'TTY:', the file name of the key board.

TYPE:

**TEST.\_\_\_\_=TTY:** and press **RETURN**

↑  
your initials.

Now let's input the data. PIP is waiting for it. Remember to type everything that is in bold and press every bold key.

TYPE:

**This is line 1** and press **RETURN** and **LINEFEED**

**This is line 2** and press **RETURN** and **LINEFEED**

**This is line 3** and press **RETURN** and **LINEFEED**

Do you remember that CTRL + Z is the 'CTRL' and the 'Z' keys pressed at the same time?

PRESS:

**CTRL + Z**

What happened? PIP responded with the '\*' again to show that the command is complete.

Why? Because the CTRL + Z command tells PIP (and the ED utility) that there is no more data. It acts as an 'End of File' for the keyboard.

PRESS:

**RETURN**

What happened? CP/M responded with the A> to show that the PIP utility is completed.

Why? Because the RETURN command tells PIP that there are no more commands for it. Let's look at the directory.

TYPE:

**DIR TEST.\*** and press **RETURN**

Do you see TEST listed in the directory?

Now let's try some commands with our test file.



## Printing A File

If you don't have a printer attached to your system you won't be able to do this section. If you do, check before you start that the printer is attached to the system, that it is turned on and that it is 'on-line'. A printer usually has some indicators that show what its status is and some buttons for setting the status.

Printing a file is just like copying a file. Do you remember the PIP command?

```
TO=FROM(parameters)
```

In this case our 'TO' is the printer. Do you remember that we can use a device name anytime we would use a file name? Do you remember the device name for the printer?

TYPE:

```
PIP          and press RETURN
```

TYPE:

```
LST:=NEWTEST.____ and press RETURN
      ^
      |
      | your initials
```

PRESS:

```
RETURN
```

If you accidentally do this without having a printer your system will lock up and you will not be able to do anything with your keyboard.

Press the reset button and after the self-test message and the '\*' appears, press RETURN. You will automatically be returned to disk 'A', USER0; therefore, you will need to return to USER1. Do you remember how to change User Directories? If not, you can look back in this Getting Started Manual (Section 2.3, Directories) for the instructions.

## Erasing A File

We really need only one copy of our test file, so let's erase the second copy. You will periodically want to erase files that are old, redundant, or not in use. Of course you will have made a copy of the ones you want

to keep by backing them up on the magnetic tape cartridge. We'll talk about that later.

Erasing a file is a drastic step; once it's gone, you can't get it back. So be sure to check your command before you press the RETURN. Let's erase the extra file.

TYPE:

```
ERA TEST2.____ and press RETURN
      ^
      |
      | your initials
```

Let's check the directory.

TYPE:

```
DIR *.____ and press RETURN
      ^
      |
      | your initials
```

You no longer have TEST2 listed in the directory so CP/M won't be able to find it for you. ERA works with wild cards in the file name so you can erase a whole set of files. For example:

```
ERA *.BAK
```

erases all files with a secondary name BAK. However, ERA works only on files in the directory you are working in, which keeps you from accidentally erasing files you have listed in other directories.

## Summary

So far we have learned to:

1. Set the date and time (DATE, TIME)
2. Move from one disk drive to another (A:, B:,...)
3. Move from one directory to another (USER 1, USER 2,...)
4. List all the files in a directory (DIR)
5. Use the wild card characters to list some of the files in a directory (DIR \*, ?)
6. Create a file using PIP (TO=TTY:)
7. Use device names with PIP (TTY:, LST:)
8. Rename a file (REN NEW=OLD)
9. Copy a file using PIP (TO=FROM)
10. Print a file using PIP (LST:=FROM)
11. Erase a file (ERA NAME)



CP/M allows files to have the same name so long as they're not in directory USER0. That's because files listed in the USER0 directory on disk A are listed in every directory on disk A. Files listed in the USER0 directory on disk B are listed in every directory on disk B. Do you remember how to tell PIP there are no more commands?

PRESS:

**RETURN**

Besides being able to copy files from one disk to another, PIP allows you to copy files from one directory to another.

Let's move to a different directory. Do you remember how?

TYPE:

**USER 2** and press **RETURN**

Now let's run PIP again.

TYPE:

**A:PIP** and press **RETURN**

Remember the PIP command form is:

**TO=FROM(parameters)**

One parameter is Gn for GET. The Gn parameter tells PIP the source file is listed in directory USERn. (G1) tells PIP to look in directory USER1 for the file.

TYPE:

**TEST4.\_\_\_\_=A:TEST3.\_\_\_\_(G1)** and press

**RETURN**

your initials   your initials

Remember we created TEST3 on disk A and we are on disk B. We had to specify disk A for the file TEST3 so that the system would know where to find it.

PRESS:

**RETURN**

There are many more parameters for PIP and many more copy options. For example, PIP recognizes the '\*' character so you can copy an entire group of files.

## 2.7 BACKING UP FILES

---

So far we have copied files from one disk to another and one directory to another. Now let's copy our files to the magnetic tape cartridge. This is an important procedure because if we should erase a file by mistake we can recover it if we have made a copy on the magnetic tape.

To back up our files we run the ARCHIVE system utility.

Insert a new unused tape cartridge in the drive (the side with the 'SAFE' label should go in first).

CP/M will make sure the tape is 'at the beginning' (positioned at the 'load point').

**NOTE:** ARCHIVE needs the DATE and TIME to be set. It will tell you and terminate itself if they are not already set. If you make a typing mistake use the BACKSPACE key to delete the character. You should insert the magnetic tape cartridge before running ARCHIVE. If you forget, be sure to do it before answering the questions ARCHIVE will ask you.

TYPE:

A:ARCHIVE and press RETURN

### Using A New Tape

ARCHIVE asks the following questions.

Was this tape formatted with the TFORM command (Y/N)? N

We answer 'N' since this is a new tape.

Number of tensioning passes required (CR = 0)? 3  
and press RETURN

We answer with '3' since this is a new tape. The tape will now do some rewinding to insure smoothness of the tape. Wait until this is complete before typing any further.

Like other system utilities ARCHIVE has a prompt, '-'. ARCHIVE also has a 'help' command.

TYPE:

HELP and press RETURN

ARCHIVE lists its commands.

To use a new tape we first need to use the TFORM command. TFORM creates a directory for the tape. The directory is used to catalog the files you will copy onto the tape so they can be found again.

TYPE:

TFORM and press RETURN

Answer the questions. You can select any name and any serial number although you probably want to number your tapes consecutively and select names that will describe the files that are being copied. For this test you could use your first name and serial number 1. Don't use spaces or periods (.) in the name.

TYPE:

Your Name and press RETURN

TYPE:

The serial number you want to use  
and press RETURN

To copy files onto the tape we use the FBACK command. ARCHIVE recognizes the '\*' character so you can back up all your files with one command.

TYPE:

FBACK \*.\* and press RETURN

ARCHIVE tells you the names of the files as it is copying them.

When all your files are copied, you can check the tape directory.

TYPE:

TDIR and press RETURN

Your files have been cataloged in the tape directory.

Now that we've copied all our files we can stop ARCHIVE.

TYPE:

**END** and press **RETURN**

**NOTE:** The **END** command writes the directory onto the tape so always use **END** when you are finished with **ARCHIVE**. If you forget you will need to redo the copy operation since the files won't be listed in the directory on the tape.

### Using A Tape With Files On It

Now let's use the tape to restore the files you just copied.

TYPE:

**A:ARCHIVE** and press **RETURN**

ARCHIVE asks:

Was this tape formatted with the **TFORM** command (Y/N)? **Y**

This time answer 'Y'. The tape name and serial number along with the date, time, and date last updated are displayed for you.

ARCHIVE asks:

Is this the correct tape (Y/N)? **Y**

This is a safety check to make sure you have the right tape.

Number of tensioning passes required (CR = 0)?

PRESS:

**RETURN**

We don't need any tensioning passes since we just used the tape; periodically, however, you will want to answer '3' as you did for a new tape.

To load files from the magnetic tape use the **FLOAD** command.

TYPE:

**FLOAD \*.\*** and press **RETURN**

What happened? **ARCHIVE** prevents you from erasing a file you might want to keep. If you type 'Y', **ARCHIVE** will write over the file you have stored on the disk.

TYPE:

**N**

ARCHIVE asks:

New file name (CR = do not load file)?

If you press **RETURN**, **ARCHIVE** will not load the file. By typing the file name

NTEST.\_\_\_\_

↑ your initials

ARCHIVE creates a new file for you with the name

NTEST.\_\_\_\_

↑ your initials

TYPE:

**NTEST.\_\_\_\_** and press **RETURN**

↑ your initials

Finish with **ARCHIVE**.

TYPE:

**END** and press **RETURN**

We have learned to format a new tape and backup and restore files. There are many more options with **ARCHIVE**; the User Reference Manual and Quick Reference Guide describe them all.

### Summary

This has been a quick introduction to **CP/M**. Try these commands and the special keys until you feel comfortable with them. Remember the User Reference Manual describes each command in detail; you can refer to it if you have a problem. Your dealer, who has a lot of experience with **CP/M**, can help you if you can't find it in the manual.

You are ready to start using your system. Good luck!

## 2.8 GLOSSARY

---

### Some Definitions

Here is a list of commonly used terms that you will find throughout this manual and the other system documentation you have received with your computer. These terms are used extensively by people who use computers. You'll be using them too because they describe operations and concepts that are basic to working with a computer.

#### ALPHABETIC

Any of the fifty-two upper and lower case letters of the alphabet.

#### ALPHANUMERIC

Any of the alphabetic, numeric, or punctuation characters such as

! ; : ' " . ? , ,

plus such special characters as

@ % ^ & \* ( ) + - = / < > \ #  
{ } [ ] ` ~ .

In general, any character which can be printed or displayed.

#### AMBIGUOUS

Used to describe a non-specific file reference—i.e., a file name that may refer to more than one file through the use of the wild card characters '\*' and '?.

#### ANSI

American National Standards Institute. An agency concerned with setting up industry-wide standards. Some languages have an ANSI standard.

#### APPLICATION PROGRAM

A program which performs a specialized set of functions such as Word Processing, Accounts Payable, etc....

#### ASCII

American Standard Code for Information Interchange, pronounced Ass-kee. A communication code set used by most computers to define all the possible characters—alphanumeric as well as control characters—that the computer can process.

#### BACKUP

The process or device used to create a copy of one or more files, usually all the files on the system, for security reasons. The device should be such that it can be removed from the system and stored in a safe place. A magnetic tape cartridge is one example of a backup device.

#### BAUD RATE

The speed at which data can be transmitted from one device to another or one computer to another in one second. Roughly speaking, dividing the baud rate by 10 gives the number of characters per second.

#### BINARY

A number system with only two digits: 0 and 1. Used by computers for calculations and for representing text data.

#### BIT

A binary digit (either a 1 or a 0); the smallest unit of data capable of being stored by a computer.

#### BOOT

The operation which causes the operating system to be loaded into main memory from the disk.

#### BUFFER

A temporary data storage area in the computer memory.

#### BYTE

One character of information (8 bits equal 1 byte, which in turn equals 1 character).

## **CARRIAGE RETURN**

The key on a typewriter which positions the print head to the left margin. There is a similar key on a terminal keyboard usually abbreviated RETURN. The symbol 'CR' is also used in documentation to indicate the use of this key.

## **CHECKSUM**

Used to verify the accuracy of data being read or written on disk or tape. A summation check is calculated by the system in which the data being read or written is summed. When the operation is complete, that sum is checked against a previously computed sum to verify that no data has been changed during the operation.

## **COMMAND**

An instruction to CP/M telling it to perform an operation for you.

## **CONCATENATE**

To combine a number of separate units together, uniting them into a single unit.

## **CPU**

The Central Processing Unit. Technically, the logic and arithmetic circuitry inside a computer. The place where data actually gets processed.

## **CRT**

Cathode Ray Tube. The 'picture tube' in a computer terminal; it looks like a small television set.

## **CURSOR**

The symbol which marks the current display position on a screen, similar to the typing guide on a typewriter. A cursor can be moved up or down, left or right. It may look like a block or an underline character. It may blink or not. These characteristics may be selected by the operator or set by the system software.

## **DATA**

Facts input to the computer for processing. They may be a purchase order number, the name of a customer, etc. The computer processes the data, stores it and creates information for you, like a purchase order, a picking list, or an invoice.

## **DEFAULT**

The devices or conditions that are assumed by CP/M when you have not specified explicitly what these devices or conditions are.

## **DEVICE**

A terminal, disk, printer or other piece of equipment attached to the computer. Sometimes referred to as a peripheral.

## **DIAGNOSTICS**

Programs which test the various parts of a computer system to uncover malfunctioning units.

## **DIRECTORY**

A list of files that are contained on the disk.

## **DISK DRIVE (PHYSICAL DISK)**

A metal platter or set of platters (like a phonograph record) with magnetic coating used to record information magnetically. Sometimes disks are referred to as rotating memory or external memory since the computer can store and retrieve data on the disk very quickly. Disk storage is the amount of information, usually measured in thousands (K or Kbytes) or millions (Mbytes or Megabytes) of characters that can be recorded on it. One 'physical' disk drive may be partitioned into several 'logical' disks, each with its own name.

## **DUMP**

The process of transferring all or part of the contents of the computer memory or a disk or a tape to another device, such as the terminal display, a printer, a tape, or a disk.

**ECHO**

When a character is typed at the keyboard, it is displayed or 'echoed' on the screen by CP/M. Some characters CP/M does not echo but performs an action instead. Others are not displayable; CP/M 'echoes' these with a series of one or more displayable characters.

**END OF FILE**

A character which signals that there is no more data in the file.

**FILE**

A collection of related information organized into records, each being a complete unit.

**HARDCOPY**

Used to describe a terminal or device with printing capability. It uses paper to make a permanent record of the data or information output to it. As opposed to a CRT terminal which has a display that leaves no permanent record.

**HEXADECIMAL**

A number system with sixteen possible digits (the decimal system has 10: 0-9). These are 0-9 and A-F. Numbers expressed in hexadecimal format usually have an 'H' appended to them.

**HOME**

The upper left hand corner of the display.

**INPUT**

The process of entering data into the computer for processing. Also the data itself. Usually data is input by an operator using a terminal keyboard.

**LOAD**

The process of transferring the operating system, a program or data from a disk or other device into the memory of a computer.

**LOGICAL DISK**

A segment of a physical disk drive organized with a directory structure so that data and program files can be stored on it. One or more logical disks may reside on a physical disk drive.

**MEDIUM**

Singular of Media. The material on which data or information is recorded. The tape within your tape cartridge is magnetic tape medium.

**MODEM**

MOdulator-DEModulator. A device used to translate data in computer format to a format suitable for transmission over telephone lines.

**NUMERIC**

Any of the digits 0-9.

**OUTPUT**

The process of sending information or data to a device for display (terminal), printing (printer) or storage (disk or magnetic tape). Also the data or information itself.

**OPERATING SYSTEM**

The instructions which control and manage all of the computer system — the CPU, the memory, and all the devices attached to it.

**PARAMETER**

A descriptor which helps define an operation to be performed. For example, a file search operation might have one parameter describing or limiting the search and another parameter describing how the search is to be done. The values of the parameter specify the operation in a particular instance.

**PERIPHERAL**

Any of the devices attached to a computer which are used to input data, store data or output data, as opposed to performing calculations on it. A terminal is used for both input and output; a disk for storage.

**PROMPT**

A symbol or message used by the operating system, utility program or application program to ask the operator for information. The prompt signals the operator the way a teacher helps a student. Each program may use a different symbol for its prompt to help the operator remember which program is currently running.

**TAB**

The process or key on a typewriter used to position the print head to a preset print position. A terminal keyboard may have a key with the same function. TAB positions, however, may be set differently by different system programs and may or may not be changed by the operator.

**UNAMBIGUOUS**

The term used to describe a specific file reference — i.e., a unique filename that refers to only one particular file.

**UTILITY (PROGRAM)**

A program which may be supplied with the operating system but separate from it. It performs some set of related functions generally needed on any system.

**WILD CARD**

A character used to represent one or more characters. CP/M uses two wild cards, '\*' and '?'. '\*' stands for 1-8 characters; '?' stands for a single character. A 'joker' is a wild card in poker; trumps are wild cards in bridge.



---

Section 3

**User Reference  
Manual**

---



## 3.1 HOW TO USE THIS MANUAL

---

This User Reference Manual is organized into the following sections:

This section describes the organization symbols and conventions used in the User Reference Manual.

Section 3.2 gives a basic overview of the system.

Section 3.3 describes the commands and utilities. These are arranged in alphabetical order.

Section 3.4 lists and defines the hardware error messages.

Section 3.5 describes other programs included in the Software Distribute Tape.

Section 3.6 contains the ASCII Standard Character Set.

Section 3.7 is a subject index for Section 3.3

### Organization

Section 3.3 uses the same format to describe each utility. The information appears under the following major headings:

#### Command Format

Shows the command format (and proper syntax) for each utility and command, and indicates whether a parameter is required, optional, or one of two possible choices.

#### Tasks Performed

Outlines the utility's function(s) and purpose.

#### Using the \_\_\_\_\_ Utility (or Command)

Demonstrates how to type in commands and responses for each function of the utility and features step by step instructions for executing all operations. Entry instructions are preceded by the word 'TYPE:', as in

TYPE:

FCOMP filename1 filename2 and  
press RETURN

or, when appropriate, by 'PRESS:' as in  
PRESS:

CTRL + C

In the 'TYPE:' sample above, the words 'filename1' and 'filename2' are used to specify the needed parameters, and are to be replaced by the actual filenames when typing in the directive. Other words that specify the parameter to be typed in are:

logical-device	offset
physical-port	system
input-output	user
date-string	destination
track	source
sector	protocol-type
count	n (representing a number)
command	baud-rate
drive (representing a number)	HH MM SS
disk (representing a name parameter	

In each case, these words are replaced by the appropriate name or number. Parameters may be typed in either upper or lower case.

The meaning of these parameters as they apply to each procedure is fully defined in the instructions for using the utility.

### Examples

Presents either the command formats for specific tasks, or an actual run of the utility, showing the console dialogue (operator prompts and responses). Whenever console dialogue is presented, **bold type** identifies all

keys pressed by the operator, as illustrated in the samples below.

### **STAT TIME.COM RETURN**

or

### **0013 CALL 110 RETURN**

In each case, data to be typed in, and use of the RETURN key, is indicated in **bold**.

#### **Error Recovery**

Lists and explains any messages resulting from the operator's error rather than a system error. Unless otherwise stated, the corrective action is to rerun the utility after checking for proper syntax. System errors, i.e. disk and tape errors, are described in Section 3.4.

#### **Points to Remember**

Summarizes key considerations about the utility, to help the user better understand its functions and limitations. If necessary and whenever possible, an alternate utility is referenced to accomplish a desired task.

#### **Symbols**

Certain symbols are used to designate the kind of data that must be typed in when directing system operations. These are:

- ( ) Brackets—designating an optional parameter. This means that the operating system or utility will execute the command or function whether or not the (parameter) in brackets is typed in. Generally, the utility performs a different function if the parameter is missing.
- < > Carats—designating a required parameter. Identifies the <parameter> that must be typed in, or the operating system or utility will not execute the operation.
- | Vertical Bar—designating a choice between two parameters. Indicates that the operating system or utility will execute the operation if either parameter1 or parameter2 is typed in. Generally, the

utility performs the function differently for one parameter than it does for the other parameter.

These symbols are for reference only, and are not to be included when typing the parameter.

#### **Conventions**

Most of the information in this manual applies to both the MP/M and the CP/M operating systems. However, some of the operating system functions apply only to CP/M. Whenever a comment in a description applies only to CP/M, it is distinguished by a gray half-tone background.

The word 'RETURN' and the phrase 'and press RETURN' indicate that the key labeled 'RETURN' (on the Onyx Terminal or Sundance keyboard) should be pressed to generate the carriage return character. Almost all commands and utilities look for the carriage return character to signal the end of input.

It's important to type in all responses and directions **exactly as they are shown in the instructions**. Since the operating system uses spaces to separate the command and the parameters, ignoring spaces will cause command errors.

The following notation is used to indicate a control character:

CTRL + X

where the CTRL key is held down while pressing the letter key (A, B, C ... X).

#### **Terminology**

This User Reference Manual may contain a number of terms that are unfamiliar to the new user. Most of the terms are explained as they appear in descriptions and instructions. The following terms appear frequently throughout this Manual and are defined in Table 3-1.

#### **Additional Resources**

The C5001 and C8001 User Guides, and the Sundance Hardware Operations Manual are recommended for additional reference, along with the Software Manuals from Digital Research.

Table 3-1 User Reference Manual Terminology

Term	Refers To
<b>warm boot or warm start</b>	The reloading of the user interface portion of CP/M into memory initiated by the CTRL key function.
<b>cold boot or cold start</b>	The loading of the operating system into memory from the disk, initiated either by pressing the hardware reset button or powering on the system.
<b>physical disk drive</b>	The physical device used to store data. It is physically organized into sectors (capable of storing 512 bytes of data) and tracks (consisting of 2 sectors).
<b>logical disk</b>	A partition of a physical disk which has been organized with a directory structure to permit the storage and manipulation of program and data files.
<b>current default disk</b>	The logical disk which is referenced in the absence of a specific disk name whenever filename is used. The current default disk always appears with the operating system prompt '>'. >
<b>ambiguous</b>	The use of '*' and '?' characters in a filename to reference a set of files rather than a single file.
<b>unambiguous</b>	A fully specified filename containing no '*' or '?'.
<b>primary name</b>	The name which identifies a particular file.
<b>secondary name or extension</b>	The generic name which identifies the type of information contained in a file.

## 3.2 USING THE SYSTEM

---

### File Names

Each file has a 'primary' name given to it when the file is created. Files are created either by system utility programs, like PIP or ED, or by application programs. There are the following limitations when naming a file.

1. A primary name can be from 1 - 8 characters, alphabetic, numeric, or \$. Names can be longer but only the first 8 characters are used and displayed. The operating system makes no distinction between upper and lower case alphabetic characters; 'd' and 'D' are equivalent.

2. No periods (.), commas (,), asterisks (\*), question marks (?) or spaces can be part of the file name as these characters have special meanings.

A file may have an optional secondary name, sometimes called an extension, as well as the primary name just described. The secondary name follows the primary name and is separated from the primary name by a period (.).

It has the same restrictions as the primary name but it is only three characters long. The secondary name is used to help describe the type of file.

File names should be unique. Two files may share a name only if they are listed in different directories.

Two files may have the same primary name as long as they have different secondary names; two files may have the same secondary name as long as the primary names are different. Some possibilities are:

XYZ	no secondary name
XYZ.TXT	two files with same primary name
XYZ.DAT	
ABC.DAT	two files with same secondary name

Some commonly used file types are:

.ASM	an 8080 assembly language source program
.BAK	a backup file
.BAS	a file containing a BASIC source program

.COM	a file containing a program ready to run
.DAT	a data file
.DOC	a document file
.DTA	a data file
.HEX	a file containing 8080 machine language
.OVL	a file containing a program overlay
.OVR	a file containing a program overlay
.PRL	a file containing a page relocatable program ready to run (MP/M only).
.PRN	a file containing a listing of an 8080 assembly language program suitable for printing
.RSP	a resident system process (MP/M only).
.SUB	a file containing operating system commands
.TXT	a file containing ASCII text
.\$\$\$	a temporary or work file

File names may be ambiguous—that is they may specify a set of files with related names. Two characters called 'wild cards' are used for this purpose. 'wild cards', are used to represent various characters (letters, digits, or symbol keys). They are:

- \* an asterisk, used to represent from one to eight questionable characters.
- ? a question mark, used to represent a single displayable character.

These characters can be typed in as a kind of 'shorthand' to avoid typing and retyping an entire filename. More importantly, they allow an operating system command or utility to be executed when the exact filename is not known or is unimportant. They also allow a single command to operate many times on a set of related files without having to specify each filename separately.

## File Status

There are four status conditions occurring in pairs which a file may have:

R/W or R/O

and

DIR or SYS

'R/W' is the normal status when a file is created. It allows the file to be modified. 'R/O' prevents a file from being altered. 'DIR' status is also the normal status for a file; the file's directory entry is available to all utilities for listing or file manipulation. Certain files, however, benefit from having the 'SYS' condition. This condition causes the utilities to ignore the file directory entry.

## Disks

A physical disk is organized as a set of logical disks. There can be as many as sixteen (16) logical disks on the system. Each has a name—A, B, C, D,...etc. Each logical disk has a directory structure which allows files to be stored and accessed on it.

## Device Names

The operating system treats all devices as if they were files. Each has a name, the physical device name. Physical device names on the Onyx system include:

TTY: keyboard  
CRT: display  
LPT: printer

Logical device names include:

CON: (console)  
LST: (printer)

Two additional logical device names can be used in PIP commands:

PRN: Serves the same purpose as LST:  
EOF: Sends an end-of-file character (CTRL + Z) at the end of all ASCII data transfers.

The physical ports to which logical device names may be assigned are:

- The Terminal Port
- The Modem Port
- The Serial Printer Port
- The Parallel Printer Port

Both physical and logical device names are recognized by the operating system. Communication with the device is effected by establishing the link between the logical device name and the physical port. Thus the console (CON) may be attached to the terminal port, the modem port or the serial printer port.

The Parallel Printer Port may be used for a parallel 'Centronics' type printer or an IBM compatible 8-inch flexible disk drive.

## Control Key Functions

CTRL key functions start, restart, or stop an operation. In some cases a CTRL key function is implemented with a specific key. These are identified below, and are always noted as CTRL + (a letter). In each case, the CTRL key is held down while pressing the letter.

CTRL + C

Stops the current user program (MP/M only).

Causes a warm boot (warm start) of CP/M and initiates reloading of the user interface portion of the operating system. You may also use this control key to stop a utility or a system operation.

CTRL + D

Causes the current user program to detach from the terminal allowing it to be used for other functions (MP/M only).

Has no effect under CP/M.

#### CTRL + H

Acts just like the BACKSPACE key. Erases the character immediately to the left of the cursor and positions the cursor there.

#### CTRL + I

Acts just like the TAB key. Moves the cursor right, to the next tab stop. The tab stops are set at every 8th column (8, 16, etc.). TABs are translated into spaces and ignored.

#### CTRL + J

Acts just like the LINEFEED key and is used by the operating system just like the RETURN key. When used by a system utility or application program, the LINEFEED key causes the cursor to move down one line.

#### CTRL + M

Acts just like the RETURN key. The operating system responds by asking for more information or by responding with the prompt '>' to show that the command has been completed. When the RETURN key is used by a system utility or application program it causes the cursor to move to the beginning of the line.

#### CTRL + P

Sends all information displayed on the screen (or typed on the keyboard) to the printer. Pressing CTRL + P a second time stops sending the displayed information to the printer.

#### CTRL + R

Redisplays the current corrected command line.

#### CTRL + S

Acts just like the NO SCROLL key on the Sundance/132 and DT/132. Stops the output operation to the display without loss of data. Pressing any key causes the output to be resumed. Unlike the NO SCROLL, this key operates on all Onyx and Sundance terminals.

#### CTRL + U

Deletes the entire line that has been typed so far. The cursor is positioned on the next line but the prompt '>' does not appear.

#### CTRL + Z

Signals end of input from the keyboard for system utility programs ED and PIP. Causes ED and PIP to terminate the operation and wait for the next command.

### 3.3 COMMANDS AND UTILITIES

---

This section describes the CP/M and MP/M utilities which are supplied by Onyx. Commands and utilities supplied by Digital Research are also included but only to describe how they are invoked and terminated. Refer to the Digital Research utility manuals and system reference manuals for descriptions of how the individual commands work.

Most of the utilities run under either operating system. However, a small number of the utilities only run under CP/M. These are distinguished by a gray half-tone background. Whenever a comment in a description applies only to CP/M, it is highlighted with the half tone.

## **ANSI Switch Sundance or Onyx Terminal from VT52 Mode to ANSI Mode**

**ANSI**

### **Command Format**

ANSI

### **Tasks Performed**

ANSI switches the Onyx terminal and Sundance from VT52 mode to ANSI mode.

### **Using the ANSI Utility**

TYPE:

ANSI

The DT/80, DT/132, Sundance/80 or Sundance/132 is put in ANSI mode.

### **Error Recovery**

None.

### **Points To Remember**

If VTSETUP has been run, the Onyx terminal and Sundance return to ANSI mode whenever a warm boot occurs. Thus it is not necessary to run ANSI unless the application using VT52 mode does not cause a warm boot.

## ARCHIVE Logical File Backup

## ARCHIVE

### Command Format

ARCHIVE {filename}

### Command Summary

HELP  
TFORM  
TDIR {filename} {LAST}  
FBACK <filename>  
FLOAD <filename> {LAST| CHOOSE n}  
TVFY  
USER <n| ?>  
END

### Tasks Performed

Copies individual files and sets of files to tape and restores them to disk at a later time, effectively giving the user an infinite amount of disk storage space.

Each backup file receives a version number, allowing the ARCHIVE tape to store several versions of the same file. Keyword parameters are available to allow manipulation of the latest version or to choose one file from among several with the same name.

### Using the ARCHIVE Utility

Before using ARCHIVE, the time and date must be set using the TOD utility on MP/M or using the TIME and DATE utilities on CP/M. Place the tape cartridge in the tape drive.

#### 1A. To run ARCHIVE interactively.

TYPE:

ARCHIVE and press RETURN

and ARCHIVE will ask for all of its input from the console.

#### 1B. To use a command file,

TYPE:

ARCHIVE filename and press RETURN

and ARCHIVE will take its input from the specified file.

In both cases, if a semi-colon ';' is the first character of the input line, ARCHIVE treats the line as a comment.

### 2. Opening Dialogue

The opening dialogue is **always** conducted interactively, even when a file name was specified on the call to ARCHIVE.

After initializing its internal tables, ARCHIVE prompts:

Was this tape formatted with TFORM command (Y/N)?

TYPE:

N

and ARCHIVE proceeds directly to the tensioning prompt. It is necessary to format the tape, using the command TFORM, before it can be used.

TYPE:

Y

and ARCHIVE reads the tape header. In the case of a write-protected tape, ARCHIVE displays the following message:

\*\*\*\*Tape is WRITE PROTECTED (SAFE)\*\*\*\*  
(TFORM & FBACK disabled)

## ARCHIVE

If the tape is not write-protected, the following message appears:

```
****Tape is WRITE ENABLED (NOT SAFE)****  
(TFORM & FBACK enabled)
```

When the label indicates the tape was written by the ARCHIVE utility, the label is displayed as follows:

```
TAPE NAME SER #   FORMATTED   LAST UPDATED  
name          n   mm/dd/yy hh:mm mm/dd/yy hh:mm
```

Is this the correct tape (Y/N)?

TYPE:

N

and ARCHIVE displays:

```
Load proper tape and type carriage return  
(or ^C to abort)
```

TYPE:

Y

and ARCHIVE proceeds to the final prompt in the opening dialogue:

```
Number of tensioning passes required (CR = 0)?
```

NOTE: Usually three tensioning passes are required when using a tape for the first time. At other times, no tensioning is necessary unless a large number of tape errors were previously encountered.

TYPE:

n and press RETURN

- n is the number of tensioning passes.

If no tensioning is required,

PRESS:

RETURN

## ARCHIVE

### 3. The Commands

ARCHIVE indicates it is ready for command input by displaying the prompt character '-'. ARCHIVE accepts commands either from the console or from the command file named in the calling sequence.

NOTE: If input is taken from a file, the commands are echoed to the console.

Each valid command and command option directs ARCHIVE to perform a particular task, as described in Table 3-2. Each command is followed by a RETURN.

### 4. FLOAD Dialogue

If the file to be loaded exists on a disk, ARCHIVE displays:

```
filename exists on disk. Overwrite (Y/N)?
```

TYPE:

Y

and ARCHIVE overwrites the existing disk file, even if it is set to R/O status.

TYPE:

N

and ARCHIVE displays:

```
New filename (CR=do not load file)?
```

PRESS:

RETURN

and ARCHIVE cancels the loading of the file at this point and continues with the next file.

A valid filename entry restores the tape copy under that name on disk (as long as it doesn't exist on disk).

**Table 3-2. ARCHIVE Commands**

<b>Task</b>	<b>Type In</b>	<b>Comments</b>
Print out this command summary.	HELP	
Format a new tape or reformat an existing ARCHIVE tape by writing a valid header and an empty directory onto track zero.	TFORM	<b>WARNING:</b> TFORM 'overwrites' a tape so that any information an old ARCHIVE tape has will be lost if it is 'TFORM'ed'.
Display the contents of the tape directory.	TDIR	
Display the directory entries of the specified file(s).	TDIR filename	
Display only the directory entries for the latest version of all files on the tape.	TDIR LAST	
Display only the directory entries for the latest version of the specified files.	TDIR filename LAST	
Backup the file(s) matching the filename parameter from disk to tape.	FBACK filename	A version number is maintained with each directory entry for easy reference to each version of a file. <b>NOTE:</b> Public files on User 0 may only be backed up from User 0. The USER command is used to move between user directories.
Load the latest version of the file(s).	FLOAD filename FLOAD filename LAST	FLOAD with filename loads the latest version of the file (LAST is assumed).
Load the specified version of the file(s).	FLOAD filename CHOOSE n	FLOAD with CHOOSE n specified loads only the file(s) matching the possibly ambiguous filename parameter and having a sequence number equal to the n parameter.
Move to any valid user directory.	USER n	
Display the current user directory.	USER ?	
Read the tape header, directory, and every data block to verify the integrity of the ARCHIVE tape.	TVFY	
Write an updated header and directory to the tape, and return to the operating system.	END	

## ARCHIVE

This prompt sequence repeats until the name of a non-existent file is entered or the option is taken to overwrite the file or to cancel loading it. During the operation, ARCHIVE displays the tape directory entry and disk file name for the currently loading file.

**NOTE:** Public files in User 0 are always 'visible' to ARCHIVE when FLOAD is loading files to other user directories.

### Examples

In the following samples of console dialogue, bold print signifies input typed by the user. If a RETURN is required, it also appears in bold. Tape Statistics and Signoff Messages are omitted after the first example.

## ARCHIVE

1. Tension and format a new tape, backup several files, and verify the tape.

### ARCHIVE RETURN

Tape Archive/Retrieval Facility. Vn 1.12 25 Mar 82  
Copyright © Johnson-Laird, Inc.

Has this tape been formatted with the TFORM  
command (Y/N)? **N**

Number of tensioning passes required (CR = 0) **3 RETURN**

#### -TFORM

New name for tape (8 chars)? **ARCHTAPE RETURN**

Tape serial number? **1 RETURN**

Format complete.

#### -FBACK ?80.com RETURN

Backing up A:M80 .COM

Backing up A:L80 .COM

Backup complete.

#### -TDIR RETURN

FILE	NAME	SEQ #	DATE	TIME	BYTES
L80	COM	1	8/06/80	15:21	10K
M80	COM	1	8/06/80	15:22	18K

There are 2 directory entries. 194 entries available.

#### -TVFY RETURN

Verify complete — no errors.

#### -END RETURN

(Tape Statistics are provided to monitor tape quality)

Error Type	( Recovery =>	Easy	Hard )
CRC Errors		0000	0000
Failed to detect data		0000	0000
Data but no sync char.		0000	0000
Waited too long for char.		0000	0000
Waited too long for CRC		0000	0000
Read-after-write data err.		0000	0000

ARCHIVE ended.

## ARCHIVE

## ARCHIVE

2. Load files from a previously entered ARCHIVE tape onto the disk in User 15. The tape has been write protected.

### ARCHIVE RETURN

Tape Archive/Retrieval Facility. Vn 1.12 25 Mar 82  
Copyright © Johnson-Laird, Inc.

Has this tape been formatted with the TFORM command (Y/N)? Y

Number of tensioning passes required (C = 0)? RETURN

\*\*\*Tape is WRITE PROTECTED (SAFE)\*\*\* (TFORM & FBACK disabled)

TAPE NAME	SER #	FORMATTED	LAST UPDATED
ARCHTAPE	1	08/06/80 14:51	08/06/80 15:7

Is this the correct tape (Y/N)? Y

### -TDIR \*.COM RETURN

FILE	NAME	SEQ #	DATE	TIME	BYTES
L80	COM	1	08/06/80	15:21	10K
M80	COM	1	08/06/80	15:22	18K

There are 2 directory entries. 194 entries available.

### -USER 15 RETURN

Current user = 15.

### -FLOAD B:M80.COM RETURN

B:M80.COM exists on disk. Overwrite (Y/N)? N  
New file name (CR = do not load file)?

### B:NEWM80.COM RETURN

Loading M80 COM 1 08/06/80 15:2218K  
Disk file name is B:NEWM80.COM  
Load complete.

### -END RETURN

## ARCHIVE

3. Take command input from a file instead of the console. The contents of the input file ARCH.INP is listed below; the actual commands are capitalized.

```
; See what's on the tape.  
TDIR *.DDS  
; Go to user 5.  
USER 5  
; Load the latest version of all .DDS files.  
FLOAD *.DDS LAST  
; End of restore operation  
END
```

Lines beginning with a semi-colon are taken as comments by ARCHIVE. The opening dialogue **must** be answered interactively by the user, even though an input file has been specified; therefore the input file contains no responses for that dialogue.

## ARCHIVE

### ARCHIVE ARCH.INP RETURN

Tape Archive/Retrieval Facility. Vn 1.12 25 Mar 82  
Copyright © Johnson-Laird, Inc.

Has this tape been formatted with the TFORM  
command (Y/N)? Y

\*\*\*\*Tape is WRITE PROTECTED (SAFE)\*\*\*\* (TFORM &  
FBACK disabled)

TAPE NAME	SER #	FORMATTED	LAST UPDATED
OLDTAPE	2	07/23/80 10:2	08/03/80 23:30

Is this the correct tape (Y/N)? Y

Number of tensioning passes required (CR = 0)? RETURN

-; See what's on the tape.

-TDIR \*.DDS

FILE	NAME	SEQ #	DATE	TIME	BYTES
CUSTMAST	DDS	1	8/01/80	20:14	2K
CUSTMAST	DDS	2	8/03/80	8:31	2K
CUSTMF	DDS	1	8/01/80	20:13	10K
TERMMF	DDS	1	8/01/80	20:13	26K

There are 19 directory entries. 177 entries available.

-; Go to user 5.

-USER 5

Current user = 5.

-; Load the latest version of all .DDS files.

-FLOAD \*.DDS LAST

Loading CUSTMAST DDS 2 8/03/80 8:31 2K

Disk file name is A:CUSTMAST DDS

Loading CUSTMF DDS 1 8/01/80 20:13 10K

Disk file name is A:CUSTMF DDS

Loading TERMMF DDS 1 8/01/80 20:13 26K

Disk file name is A:TERMMF DDS

Load complete.

-; End of restore operation.

-END

# ARCHIVE

## Error Recovery

1. If the tape header cannot be verified, ARCHIVE displays:

Unrecognizable tape. Proceed (Y/N)?

If the user responds 'N', ARCHIVE will prompt:

Load proper tape and type carriage return  
(or ^C to abort)

At this point the user may change tapes or terminate ARCHIVE.

PRESS:

CTRL + C

to terminate ARCHIVE.

**WARNING:** Do not change tapes at any other point while using the ARCHIVE utility.

2. TDIR cannot be used on an unformatted tape. If the tape header cannot be read, ARCHIVE displays the following message:

Tape must be formatted before proceeding.

**WARNING:** Attempting to read a tape that has never been formatted will cause unpredictable results.

3. When using TVFY, a bad block in the header or directory is very often fatal and means that files cannot be loaded from the tape. However, at the end of processing, ARCHIVE will rewrite these areas if the tape has been altered (by the FBACK or TFORM commands). Thus, it is possible that a 'bad' tape can be made 'good' again. A bad block in a file means that file alone cannot be recovered. If possible, take another backup of the file.

# ARCHIVE

**NOTE:** For this command **only**, ARCHIVE does not terminate if a fatal tape error occurs. It is therefore possible for the utility to 'hang' in an endless loop, attempting to read a bad block. If this occurs,

PRESS:

CTRL + C

(hold down CTRL while pressing C) to terminate ARCHIVE. However, if ARCHIVE is terminated in this manner, the tape header and directory are not rewritten, and any files backed up in the current run are lost.

## Points To Remember

The ARCHIVE tape directory is kept in main memory during processing and, accordingly, is limited to 196 files.

**WARNING:** The END command writes the copy of the tape directory kept in main memory onto the tape. If the directory has been modified and if ARCHIVE is terminated without using the END command, the tape directory will not include any files which have been copied to tape during the session.

If there are many files to be backed up (which are to be reloaded subsequently), instead of using FBACK \*.\* (which is very quick), it is worth taking the time to back-up the files in (approximate) alphabetical order, e.g. — FBACK A\*.\*, FBACK B\*.\*, etc.

You can use the FILESET utility to create alphabetized lists of filenames.

**Command Format**

ASM <filename>

**Tasks Performed**

Produces a machine language version of an 8080 assembly language program.

**Using the ASM Utility**

TYPE:

ASM filename      and press RETURN

- filename is the primary name of a source file containing assembly language statements. The file's secondary name is assumed to be ASM. The filename may be preceded by an optional disk name.

ASM creates two files:

- filename .PRN which contains the program source statements along with the machine code and any error messages. This file may be displayed on the console or printed using the PIP utility.
- filename .HEX contains the 8080 machine language code in a format suitable for the LOAD utility.

**Example**

The following is a sample of a command format.

Assemble a file TEST.ASM.

ASM TEST

To print the file use the PIP utility

PIP LST:=TEST.PRN

Use the LOAD utility to create an executable program.

LOAD TEST

**Error Recovery**

If the source file does not exist, ASM displays:

NO SOURCE FILE PRESENT

NOTE: Refer to the CP/M Assembler (ASM) User's Guide or the MP/M Programmer's Guide for program error messages.

**Points To Remember**

If the source file is on a disk other than the current one, the .PRN and .HEX files will be stored on that disk also.

## Command Formats

ASSIGN  
ASSIGN <logical-device> <input-output> {physical-port}

## Tasks Performed

Establishes connections between 'logical' devices and 'physical' ports so data can be read from and written to physical devices using logical device names. It can also display the current device assignments.

## Using the ASSIGN Utility

1. To display ASSIGN parameters,

TYPE:

ASSIGN and press RETURN

ASSIGN displays a description of all the parameters.

2. To connect a logical device name to a physical port,

TYPE:

ASSIGN logical-device input-output physical-port  
and press RETURN

- logical-device is one of the following:
  - CONSOLE or TERMINAL
  - LIST or PRINTER
- input-output is one of the following:
  - INPUT
  - OUTPUT

- physical-port is one or more of the following:
  - CONSOLE or TERMINAL (the terminal port)
  - MODEM (the modem port)
  - SERIAL (the serial printer port)
  - PARALLEL (the parallel printer port)

NOTE: Device names, port names, and the input-output parameter may be abbreviated to the first character and entered in either upper or lower case (i.e., C for console, P for printer).

3. To display the current assignment for a logical device,

TYPE:

ASSIGN logical-device input-output

ASSIGN displays the current assignment for the specified logical device and I/O direction (either Input or Output).

The **output** of a logical device can be ASSIGNED to more than one physical port. See the 'Remote Console' example that follows.

**Examples**

The following are samples of command format.

1. Assign the logical printer to the serial printer port.

```
ASSIGN LIST OUTPUT SERIAL or ASSIGN L O S  
ASSIGN LIST INPUT SERIAL or ASSIGN L I S
```

2. Assign the logical printer to the parallel printer port.

```
ASSIGN LIST OUTPUT PARALLEL or ASSIGN L O P
```

3. Use a remote Console by assigning input and output to a remote terminal. This permits access to the Onyx system via the modem port which is connected to a phone line with a modem.

At the local system console assign logical device output to both the modem and terminal ports.

```
ASSIGN CONSOLE OUTPUT MODEM CONSOLE or  
ASSIGN C O M C
```

Assign the input to the modem port.

```
ASSIGN CONSOLE INPUT MODEM or ASSIGN C I M
```

From this point on, console input is read from the modem port. Output intended for the console goes to both the modem port and the local terminal allowing the local user to monitor the progress of the operation.

If the work entered on a remote terminal is proprietary in nature, the remote user prevents the confidential output from appearing on the local terminal by removing the console output assignment to the terminal as follows:

```
ASSIGN CONSOLE OUTPUT MODEM or  
ASSIGN C O M
```

This directs console output to the only modem port and not to the local terminal port.

Control of the system is restored to the local console by the reassigning of the logical console input and output at the remote terminal as follows:

```
ASSIGN CONSOLE OUTPUT CONSOLE or  
ASSIGN C O C  
ASSIGN CONSOLE INPUT CONSOLE or ASSIGN C I C
```

**Error Recovery**

There are error messages for:

- Invalid logical device name
- Illegal input/output direction
- Illegal physical device

Valid options are displayed following the error message.

**Points To Remember**

Assignments remain in effect only until the next system reset or start-up. The FREEZE utility will make any assignments permanent and remove the need to rerun ASSIGN.

**Command Format**

CCREADY

**Tasks Performed**

Permits use of the 'type-ahead' feature afforded by interrupt-driven input drivers.

A type-ahead occurs when running a program that recognizes certain control characters entered from the keyboard. For this reason the program continuously monitors keyboard status, checking for entry of one of these control characters. As a result, any characters that are not one of these special characters are discarded.

The CCREADY program conditions the input drivers to respond only to the entry of a CTRL + C character. This allows the system to accept all other data characters and place them in the terminal's type-ahead buffer.

**Using the CCREADY Utility**

TYPE:

CCREADY and press RETURN

**Example**

The following is a sample of using CCREADY.

```
DO CCREADY\MBASIC  
PAYROLL\CCREADY\MBASIC FEDTAX RETURN
```

In the example above, if PAYROLL automatically chains into FEDTAX without exiting from MBASIC, then the command line would read:

```
DO CCREADY\MBASIC PAYROLL RETURN
```

That is, there would be no need to call CCREADY a second time.

**Error Recovery**

None needed.

**Points To Remember**

1. The effect of invoking the CCREADY program lasts only until the program terminates (the next warm-boot) or, in the case of Microsoft BASIC, until the 'SYSTEM' command is entered to return to the operating system.
2. Certain other products, such as MicroPro's WordMaster™ and WordStar™ will not function properly if they are called immediately after CCREADY has been called.

™Word Master and Wordstar are trademarks of MicroPro International

## **COL80/COL132** Switch Sundance/132 and Onyx DT/132 Terminals between 80 and 132 column mode

## **COL80/COL132**

### **Command Formats**

COL80  
COL132

### **Tasks Performed**

COL132 switches the 132 column Onyx terminal and 132 column Sundance from 80 column mode to 132 column mode. It has no effect on the 80 column versions. COL80 returns the Onyx terminal and Sundance to 80 column mode.

### **Using the COL80 and COL132 Utilities**

1. To enable 132 column mode,

TYPE:

COL132      and press RETURN

Applications which use only 80 columns will continue to work properly in 132 column mode. Applications which expect 132 column mode need to have COL132 run prior to executing them.

2. To disable 132 column mode (return to 80 column mode),

TYPE:

COL80      and press RETURN

### **Error Recovery**

None needed.

### **Points To Remember**

If VTSETUP has been run, the Onyx terminal and Sundance return to 80 column mode whenever a warm boot occurs.

## **DATE** Set/Display System Date

## **DATE**

### **Command Format**

DATE {date-string}

### **Tasks Performed**

Sets or displays the current date. This date remains unchanged as long as the system is not reset and is accessible to application programs.

### **Using the DATE Utility**

1. To set the system date,

TYPE:

DATE date-string and press RETURN

No check is made on the format or the validity of the date string. The date string is terminated by the first blank or the end of the line; to enter the date in the form 1 May 80, type 1-May-80.

2. To display the current 'date',

TYPE:

DATE and press RETURN

If no date has been entered since system start-up, the system displays the current operating system release date.

3. Accessing the Date from a Program

The date string parameter is stored in upper memory starting at location 0FFDDH (65501 decimal). The string is terminated by a byte containing 00H.

**NOTE:** Since DATE allows a 10 character string to be entered from the console and saves the string in high memory, for subsequent display or user program access, this facility may be used for purposes other than the calendar date.

### **Example**

The following is a sample of command format.

Set the current date.

DATE 4/20/82

### **Error Recovery**

Any 10 character alphanumeric string is acceptable. If an invalid character (such as CTRL + Y) is entered, DATE displays the current legal string characters followed by a '?'.  
DATE 4/20/82?

### **Points To Remember**

The date is not altered by the system clock. It must be entered each day.

## Command Format

DBACK

## Tasks Performed

Copies one or more logical disks onto a backup tape. The user specifies each logical disk which is to be included in the backup.

## Using the DBACK Utility

Before using DBACK, the time and date must be set using the TIME and DATE utilities. After placing the tape cartridge in the tape drive,

TYPE:

DBACK and press RETURN

DBACK asks for the required tensioning passes as follows:

Enter number of tape tensioning passes (CR = 0)

NOTE: Usually three tensioning passes are required when using a tape for the first time. At other times, no tensioning is necessary unless a large number of tape errors were previously encountered.

TYPE:

n and press RETURN

- n is the number of passes to be made.

PRESS:

RETURN

If no tensioning is needed.

DBACK completes the dialogue before performing any tensioning to allow the system to be left unattended while the load is proceeding.

DBACK next prompts for the logical disks selected for backup, beginning with disk A, as follows:

Enter single character 'Y' to backup given disk  
Backup disk A - y?

TYPE:

Y

and DBACK proceeds to the next logical disk.

Backup disk A - y? Yes  
Backup disk B - y?

NOTE: DBACK adds "es" to the typed-in 'Y'.

TYPE:

N

and DBACK proceeds to the next logical disk.

N <=== Not Backed Up!

NOTE: DBACK adds <=== Not Backed Up! to a 'N' response.

The DBACK prompting sequence continues until all disks have been addressed, responding to a typed-in 'Y' or 'N' for each query.

If more data than will fit on a standard DC300A tape is to be backed up, DBACK prompts with the following message to verify that an extra-long tape is in use:

CAUTION: Confirm that you are using a DC300XL tape cartridge.

Enter a carriage return to continue or Ctl-C to abort.

## DBACK

To respond:

PRESS:

RETURN

only if an extra-long tape is in use. If the tape is too short, the backup will fail when it reaches the end of tape.

PRESS:

CTRL + C

(hold down CTRL while pressing C) if the standard DC300A tape is in use. If the user has two physical disk drives, DBACK will prompt to backup the second drive (logical D:, E:, and F:) as well. When the total requested dump nears the absolute limit of a DC300XL tape (about 12 Mbytes), DBACK prints the following warning message:

Caution — a DC300XL cartridge can only hold approx. 25,000 512 Byte sectors.

Currently the total you have requested is 21,487.

NOTE: The number of sectors requested may not equal the amount in the example. Requesting more logical disks at this point will cause DBACK to fail when the end of tape is reached.

### Example

The following is a sample of console dialogue.

Backup disk B.

#### DBACK RETURN

DBACK v1.20 17 Apr 82

Enter number of tape tensioning passes (CR if 0) RETURN

Enter single character 'y' to backup given disk

Backup disk A - y? N <=== Not Backed Up!

Backup disk B - y? Yes

Label is: B 4/20/82 4:16. Number of sectors 6540.

Backing up disk B

## DBACK

### Error Recovery

If the time and date have not been set, DBACK will immediately print the following message(s) and return to the operating system.

Please set time using 'time hh mm ss'

Please set date using 'date mm/dd/yy'

Refer to the section on Hardware Error Messages (Section 3.4) for interpreting tape errors.

### Points To Remember

Since DBACK copies a disk on a sector by sector basis, it is impossible to retrieve individual files from a DBACK tape. Use the ARCHIVE utility to save and restore disks on a file by file basis.

There should be only one user on the system when running DBACK under MP/M.

## Command Format

DDT  
DDT <filename>

## Tasks Performed

Permits interactive testing and debugging of programs written in 8080 assembly language.

## Using the DDT Utility

1A. To start DDT,

TYPE:

DDT and press RETURN

1B. To start DDT and load the program to be tested,

TYPE:

DDT filename .HEX or DDT filename .COM  
and press RETURN

- filename is the .HEX or .COM program to be tested and may be preceded by an optional disk name.

DDT issues its prompt '-' and waits for commands.

**NOTE:** All DDT commands are fully described in both CP/M and MP/M II manuals from Digital Research. For CP/M systems refer to Section 4, titled "CP/M Dynamic Debugging Tool", of the CP/M Operating System Manual. For MP/M II systems refer to Section 5, titled "RDT", of the MP/M II Operating System Programmer's Guide.

2. To save an altered version of the file,

PRESS:

CTRL + C

(hold down CTRL while pressing C), then use the SAVE command.

3. To terminate DDT,

PRESS:

CTRL + C

## Example

Refer to the DDT User's Guide for an extensive example of using DDT commands.

## Error Recovery

If DDT cannot find the file, it responds with a ?

**NOTE:** Refer to the DDT User's Guide for command error processing.

## DDUMP Dump Physical Disk Sectors

## DDUMP

### Command Format

DDUMP <track> <sector> <count>

### Tasks Performed

Displays physical (512 byte) disk sectors in hexadecimal and ASCII formats.

### Using the DDUMP Utility

TYPE:

DDUMP track sector count and press RETURN

- track is a CP/M 'track' from which to begin dumping.
- sector is a disk physical (512-byte) sector within the requested track. If sector is greater than the total number of sectors on the track, the track is incremented and the sector decremented by the number of sectors per track until the system arrives at a valid sector number.
- count is the total number of 512-byte sectors to dump.

NOTE: A 'track' is equivalent to two 512-byte sectors. To determine the track and sector values needed by DDUMP, divide the absolute sector address by 2. The result is the track; the remainder (either 0 or 1) is the sector. However, if only the absolute sector is entered with track =0, DDUMP calculates the track and adjusts the sector value.

Table 3-3 provides useful data about the physical/logical disks on the Onyx systems:

Table 3-3. Physical/Logical Track Assignments

	SU Systems		MU Systems	
	Track	Sector	Track	Sector
For the 6MB drive -				
Start of Directory A:	150	0	150	0
Start of Directory B:	2490	0	2490	0
For the 10MB drive -				
Start of Directory A:	150	0	150	0
Start of Directory B:	2490	0	2490	0
Start of Directory C:	6024	0	6540	0
For the 20MB drive -				
Start of Directory A:	150	0	150	0
Start of Directory B:	2490	0	2490	0
Start of Directory C:	9975	0	10680	0
For the 40MB drive -				
Start of Directory A:	150	0	150	0
Start of Directory B:	2490	0	2490	0
Start of Directory C:	9975	0	10680	0
Start of Directory D:	17460	0	18870	0
Start of Directory E:	19800	0	21210	0
Start of Directory F:	27285	0	29400	0

# DDUMP

# DDUMP

## Example

The following are samples of command formats.

Examine the contents of absolute sector 517; divide 517 by 2.

```
DDUMP 258 1 1
```

Have DDUMP calculate the track number.

```
DDUMP 0 517 1
```

## Error Recovery

If one or more parameters are missing, DDUMP displays:

```
Parameter missing.
```

```
Expects absolute trk sec cnt.
```

## Points to Remember

DDUMP works on the internal disk drive only.

DDUMP works with physical sectors on the disk rather than with files. To examine the contents of a file use the FDUMP utility.

**DIR Display Directory Contents****Command Format**

DIR {filename}

**Tasks Performed**

Displays the names of all files in the current user directory which satisfy the possibly ambiguous filename. Files in User 0 which have 'DIR' status are also shown if PUB has been run.

**Using the DIR Command**

1. To display all the file names in the current directory,  
TYPE:

DIR and press RETURN

DIR lists the files on the current default disk in the current user directory.

2. To display a set of file names in the current directory,  
TYPE:

DIR filename and press RETURN

- filename is a possibly ambiguous file name optionally preceded by a disk name.

When the disk is specified, the selected disk is addressed before the directory search takes place.

**Examples**

The following are command formats that cause DIR to search the current default disk.

```
DIR ACCT.DAT
DIR TEST?.ASM
DIR ??OAS.CAT
DIR *.COM
```

The following DIR commands cause the selected disk to be addressed before the directory search takes place.

```
DIR B:
DIR B:SC.*
DIR C:*.BAK
```

**Error Recovery**

If no files can be found which satisfy the directory request, then the message

NO FILE

is displayed.

If all files which satisfy the directory request have 'SYS' status, DIR returns to the operating system with no message.

**Points To Remember**

DIR works only within the current user directory. However, files which are in User 0 appear in the directory listing if they have been made PUBLIC and have directory status. Use the STAT utility or the INVIS utility to change file status to 'SYS'.

## **DLOAD** Logical Disk Load from Tape

## **DLOAD**

### **Command Format**

DLOAD

### **Tasks Performed**

Loads one or more logical disk images from a backup tape (written via the DBACK utility) onto one or more disks. DLOAD permits the selection of any or all logical disks to be loaded from the tape.

### **Using the DLOAD Utility**

Insert the tape cartridge in the tape drive.

TYPE:

DLOAD and press RETURN

DLOAD first attempts to read the tape label. If the tape was written by DBACK, the label is displayed in the form:

Label = X, 4/20/82 Time = 20:14, # Sectors = 12114  
Enter Control-C to abort, anything else to continue with load.

- X is the combination of all the names of the disks that are contained on the tape followed by the date and time it was created.
- Sectors is the total number of sectors stored on the tape.

If this is not the correct tape,

PRESS:

CTRL + C.

(hold down CTRL while pressing C) to terminate DLOAD.

Otherwise,

PRESS:

RETURN

to proceed to the next prompt;

Enter the number of tape tensioning passes (CR = 0)

NOTE: Usually three tensioning passes are required when using a tape for the first time. At other times, no tensioning is necessary unless a large number of tape errors were previously encountered.

PRESS:

RETURN

if no tensioning is required.

Otherwise,

TYPE:

n and press RETURN

- n is the number of passes to be made.

DLOAD completes the dialogue before performing any tensioning to allow the system to be left unattended while the load is proceeding.

## DLOAD

Next, DLOAD prompts for identification of the disks on tape to be reloaded. It is not necessary to reload all the disks originally backed up.

Enter single character response to disk selection

Load disk A - y?

TYPE:

Y

and DLOAD proceeds to the next logical disk.

Load disk A - y? Yes

Load disk C - Y?

NOTE: The 'es' characters after the 'Y' are provided by DLOAD.

TYPE:

N

and DLOAD proceeds to the next logical disk.

N <=== Not to be Loaded!

NOTE: DLOAD adds <=== Not to be Loaded! to a 'N' response.

The DLOAD prompting sequence continues until all disks have been addressed, responding to a typed-in 'Y' or 'N' for each query.

Upon completion of this dialogue, the tensioning (if requested) and the load proceed automatically.

## DLOAD

### Example

The following is a sample of console dialogue.

Load disk B from a tape created by the DBACK utility.

### DLOAD RETURN

DLOAD v1.12 26 Mar 82

Label = B 4/20/82, Time = 4:16, # Sectors = 6540

Enter Control-C to abort, anything else to continue with load RETURN

Enter number of tape tensioning passes (CR if 0) RETURN

Enter single character responses to disk selection

Load disk B - y? Yes

Loading disk B...

### Error Recovery

If DLOAD is used with a tape created by a utility from an older release, it displays the following message:

Tape Written by Housekeeper #A or #B  
Run HLOAD to load this tape.

and exits to the operating system.

Refer to the section on Hardware Error Messages (Section 3.4) for interpreting tape errors.

### Points To Remember

DLOAD loads the entire logical disk on a sector by sector basis. It is not possible to retrieve individual files from a DBACK tape. Use the ARCHIVE utility to save and restore disks on a file by file basis.

Since DLOAD writes over an entire logical disk, it will erase all files currently stored on it. Use the ARCHIVE utility to save the files that need to be kept; they can be restored to the disk when DLOAD completes.

There should be only one user on the system when running DLOAD under MP/M.

**Command Format**

DO <COMMAND 1> {\COMMAND 2} {\COMMAND 3}....(etc.)

**Tasks Performed**

DO enables entry of several commands on a single line. These commands are stored in memory and passed to the operating system or any application program when it asks for console, modem or serial printer port input.

**Using the DO Utility**

TYPE:

DO COMMAND 1\COMMAND 2\COMMAND 3....  
and press RETURN

- Command 1 is the first command,
- Command 2 is the second command,  
etc.

The backslash '\' is used to separate the commands. The vertical bar '|' is used instead of the CTRL + C character.

**Example**

List out the directories of disk B for Users 0, 1 and 3, when the current default disk is C:. A submit file cannot be used because as soon as the USER command is given, the SUBMIT file itself is not available.

Command format is as follows:

DO USER 0\DIR B:\USER 1\DIR B:\USER 3\DIR B:

The backslashes '\' are converted into carriage returns (RETURN). The commands will be passed to the operating system at the appropriate time.

**Error Recovery**

DO echoes any invalid command followed by a '?' and then terminates.

If the wrong character is used as the command separator, DO echoes the character followed by a '?'

**Points To Remember**

The operating system converts the command tail (i.e., everything but the word DO) into UPPER case. The command tail is stored in a 'forced input' buffer. This allows a character to be retrieved from the input buffer whenever there is no incoming data and the system, or an application program, has requested data from any of the physical devices.

A DO command line only functions when the operating system or an application program actually requests console input; therefore, DO commands embedded in a submit file are not executed unless the DO is the last command of the submit file.

The DO command does not function with programs such as Microsoft BASIC (or an application program running under this BASIC) unless CCREADY is called immediately prior to the Microsoft BASIC call. This enables DO to function correctly. Refer to the description of CCREADY.

The command separator character and the character used for CTRL + C may be changed with the OPTIONS utility.

## DSTAT Disk Usage Display

## DSTAT

### Command Format

DSTAT {disk}

### Tasks Performed

Displays the percentage of total disk space which is allocated to files for each user directory.

### Using the DSTAT Utility

1. To display the usage of all disks,

TYPE:

DSTAT and press RETURN

DSTAT determines usage statistics for all disks on the system.

2. To display the usage of a particular disk,

TYPE:

DSTAT disk and press RETURN

- disk is the disk name (A, B, etc.).

DSTAT provides statistics only for that logical disk.

### Example

The following is a sample of console dialogue.

Display the usage of a system with disks A, B, C configured.

### DSTAT RETURN

Figures are approximate per cent of total disk

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Free	Used
A:	51		1		9				1	1		12					25	75
B:	83		1	1	1				9							1	5	93
C:	21					17							3		1		56	42

### Error Recovery

If the requested disk is not configured, DSTAT displays:

Invalid drive requested.

and terminates.

### Points To Remember

Since integer arithmetic is used, the percentages represent approximations only. In all probability, the individual numbers will not add up to exactly 100. Also, if any files exist in a user directory, at least 1 percent will be printed for that user, even though the actual percentage may be less.

# DUMP File Dump in Hexadecimal Format

# DUMP

## Command Format

DUMP <filename>

## Tasks Performed

Dumps the contents of a file in hexadecimal format.

## Error Recovery

If the file cannot be found, DUMP displays:

NO INPUT FILE PRESENT ON DISK

## Using the DUMP Utility

TYPE:

DUMP filename and press RETURN

- filename is an unambiguous file reference.

DUMP displays the contents of the file in hexadecimal format. The leftmost column indicates the byte count (hexadecimal).

## Example

Dump the contents of the file TEST. The console dialogue is as follows:

### DUMP TEST RETURN

```
0000 21 00 00 39 22 15 02 31 57 02 CD C1 01 FE FF C2
0010 1B 01 11 F3 01 CD 9C 01 C3 51 01 3E 80 32 13 02
0020 21 00 00 E5 CD A2 01 E1 DA 51 01 47 7D E6 0F C2
0030 44 01 CD 72 01 CD 59 01 0F DA 51 01 7C CD 8F 01
0040 7D CD 8F 01 23 3E 20 CD 65 01 78 CD 8F 01 C3 23
0050 01 CD 72 01 2A 15 02 F9 C9 E5 D5 C5 0E 0B CD 05
0060 00 C1 D1 E1 C9 E5 D5 C5 0E 02 5F CD 05 00 C1 D1
0070 E1 C9 3E 0D CD 65 01 3E 0A CD 65 01 C9 E6 0F FE
0080 0A D2 89 01 C6 30 C3 8B 01 C6 37 CD 65 01 C9 F5
0090 0F 0F 0F 0F CD 7D 01 F1 CD 7D 01 C9 0E 09 CD 05
00A0 00 C9 3A 13 02 FE 80 C2 B3 01 CD CE 01 B7 CA B3
00B0 01 37 C9 5F 16 00 3C 32 13 02 21 80 00 19 7E B7
00C0 C9 AF 32 7C 00 11 5C 00 0E 0F CD 05 00 C9 E5 D5
00D0 C5 11 5C 00 0E 14 CD 05 00 C1 D1 E1 C9 46 49 4C
00E0 45 20 44 55 4D 50 20 56 45 52 53 49 4F 4E 20 31
```

and so on...

## Command Format

DVFY

## Tasks Performed

Verifies the data on a backup tape written with the DBACK utility is readable.

## Using the DVFY Utility

Insert the tape cartridge in the tape drive.

TYPE:

DVFY and press RETURN

DVFY attempts to read the tape label and displays the following message:

Label = X, 4/20/80, Time = 20:14, # Sectors = 12114  
Enter Control-C to abort, anything else to continue verification.

- X is the combination of all the names of the disks that are contained on the tape followed by the date and time it was created.
- Sectors is the total number of sectors stored on the tape.

If this is not the correct tape,

PRESS:

CTRL + C

(hold down CTRL while pressing C) to terminate DVFY.

Otherwise,

PRESS:

RETURN

to proceed to the next prompt:

Enter number of tape tensioning passes (CR if 0)

Normally, no tensioning passes are required, unless tape errors caused failure of a previous DVFY or DLOAD.

PRESS:

RETURN

if no tensioning is required.

Otherwise,

TYPE:

n and press RETURN

- n is the number of tensioning passes.

DVFY completes the dialogue before performing any tensioning to allow the system to be left unattended while verification is proceeding.

Once the operation is set up, verification (and tensioning, if any) proceeds automatically.

## DVFY

## DVFY

### Example

Verify a tape created by DBACK that backed up disk B.  
The console dialogue is as follows:

#### DVFY RETURN

DVFY v1.12 26 Mar 82

Label = B 4/20/82, Time = 4:16, # Sectors = 6540

Enter Control-C to abort, anything else to  
continue verification RETURN

Enter number of tape tensioning passes (CR if 0) RETURN

Verifying...

Verification complete.

### Error Recovery

DVFY attempts to read the entire tape while reporting all errors encountered. Refer to the section on Hardware Error Messages (Section 3.4) for a description of possible problems encountered.

### Points To Remember

DVFY works only on a tape created by DBACK. TVFY (called from within the ARCHIVE utility), is used to verify an ARCHIVE tape. See the ARCHIVE utility.

**ED Create/Edit ASCII File****ED****Command Format**

ED &lt;filename&gt; {disk}

**Tasks Performed**

Creates and alters ASCII files which are organized into lines, each line ending with a carriage return and linefeed.

**Using the ED Utility**

1. To create or edit a file on the current disk,

TYPE:

ED filename and press RETURN

- filename may be an existent or non-existent file and may be preceded by an optional disk name.

ED creates a work file using the file's primary name and a secondary name .\$\$\$\$. When ED terminates, the work file is named filename and the original file (if any) is given the file's primary name and a .BAK secondary name. All files are stored on the same disk as the original file.

2. To edit a file stored on one disk and store the new version on another disk,

TYPE:

ED filename disk and press RETURN

- disk is the name (A:, B:, C:...) of the disk on which to store the new version of the file.

ED creates the work file on the specified disk. When ED terminates, the latest version of the file will be on the specified disk and the .BAK file on the original disk. When ED terminates, the current disk is changed to the disk specified in the command.

**Examples**

The following are samples of command formats.

1. Create a file PROG.ASM on disk B.

ED B:PROG.ASM

2. Edit a file TEST.DAT on disk B and put the new version on disk A.

ED B:TEST.DAT A:

When ED terminates the current disk is changed to A, with TEST.DAT on A: and TEST.BAK on B:.

NOTE: The commands available in the editor are fully documented in the MP/M User's Guide.

The commands available in the editor are fully documented in the User Guide 'ED: a Context Editor for the CP/M Disk System.

**Error Recovery**

If a file already exists on the target disk, ED displays:

FILE EXISTS

and terminates. The file must be erased or renamed before ED can be restarted.

If a file has a status of read only, ED displays:

\*\*FILE IS READ/ONLY\*\*

and terminates.

If a file has system status, ED displays:

"SYSTEM" FILE NOT ACCESSIBLE

and terminates. The file's status must be changed (see the STAT or VIS utilities) before ED can be used with either file.

NOTE: Command error messages are described in the documentation listed above.

**Points To Remember**

If the .BAK file already exists, ED will delete it when it renames the original file with a .BAK secondary name. To avoid losing it, use the REN command to rename it.

An edit session can be interrupted and terminated at any time without loss of the original file. Any changes made will be lost and no .BAK file will have been created in this case.

**Command Format**

ERA <filename>

**Tasks Performed**

Removes specified files from the current user directory and makes the disk space available for reuse.

**Using the ERA Command**

TYPE:

ERA filename and press RETURN

- filename is a possibly ambiguous filename optionally preceded by a disk name.

**Examples**

The following are sample command formats. Each is entered followed by the RETURN.

1. Erase file SFTPRC.DOC from the current default disk.

ERA SFTPRC.DOC

2. Erase all files with primary name NEWCOMP in the current user directory.

ERA NEWCOMP.\*

3. Erase all files with secondary name .BAK from disk B in the user directory.

ERA B:\*.BAK

4. Erase all files on the current disk in the current user directory which satisfy the ambiguous reference TEST?.COM.

ERA TEST?.COM

5. Erase all files in the current user directory on the current disk.

ERA \*.\*

In this case the operating system prompts the console with the message:

ALL FILES (Y/N)?

which requires a 'Y' response before files are actually removed.

**Error Recovery**

If the file cannot be found, ERA displays:

NO FILE

**Points To Remember**

ERA works only within the current user directory. However, it is possible to ERASE files in User 0 if they have 'DIR' status and have been made PUBLIC. If two files bear the same name (one in User 0 and one in the current directory) and the files in User 0 are PUBLIC, ERA erases **both** files. Before initiating ERA, such files in User 0 should have their status set to 'R/O' or the PUBLIC option should be disabled using the PRIV utility.

## **ERASEALL** Erase Logical Disk

### **Command Format**

ERASEALL

### **Tasks Performed**

Erases all files, in all user directories, on the specified logical disk, eliminating the need for a USER and ERA \*. \* command sequence. ERASEALL is also used to initialize the logical disks on an external disk drive.

### **Using the ERASEALL Utility**

To erase all files on a particular disk,

TYPE:

ERASEALL and press RETURN

The program then prompts:

Enter Logical Disk for which directory is to be erased:

If the response is a valid disk name (A, B, C...), ERASEALL displays:

About to erase all files on drive X  
Enter 'y' to proceed (anything else to restart)

- X is the name of the disk

and returns to the operating system.

TYPE:

Y

and ERASEALL completes the word by printing 'es' and immediately erases all files on the disk.

TYPE:

N

and ERASEALL responds with:

<=== Not Erased!

and again requests a disk name.

## **ERASEALL**

PRESS:

CTRL + C

to return to the operating system.

Finally, ERASEALL displays:

All files on drive X erased.

### **Example**

Erase all files on disk B. The console dialogue is as follows:

#### **ERASEALL RETURN**

ERASEALL v1.1 26 Mar 82 1v1.11

Enter Logical Disk for which directory is to be erased **B**  
About to erase all files on drive B  
Enter 'y' to proceed (anything else to restart) Yes  
All files on drive B erased

### **Error Recovery**

If the requested disk does not exist, ERASEALL displays:

X is an illegal disk.

and repeats the original prompt.

### **Points To Remember**

When running ERASEALL under MP/M, there should be only one user on the system.

# FCOMP File Comparison

# FCOMP

## Command Format

FCOMP <filename1> <filename2>

## Tasks Performed

Performs a byte by byte comparison of the contents of two files and displays the records which differ in hexadecimal format.

## Using the FCOMP Utility

TYPE:

FCOMP filename1 filename2 and press RETURN

- filename1 and filename2 are unambiguous file references.

If the files are successfully located, FCOMP begins the comparison. Whenever FCOMP encounters a byte within a record (128 byte sector) that differs between the two files, it displays the corresponding records. Circumflex characters ( ^ ) beneath the hexadecimal bytes indicate values which differ in the two files. The leftmost column indicates the byte count (decimal).

## Example

Compare two versions of a file TEST and TEST.BAK. The following is a sample of console dialogue.

FCOMP TEST TEST.BAK RETURN

Comparing files TEST and TEST.BAK

Sector 1 (Extent 0 / Record 1)

	File = TEST									File = TEST.BAK								
0	:	2e	73	68	31	53	50	45	45	:	2e	73	68	30	53	10	45	45
					^^		^^											
8	:	44	20	2d	20	42	61	74	19	:	44	01	0d	0a	41	44	61	7f
			^^	^^	^^	^^	^^	^^	^^									
16	:	20	21	20	20	41	41	20	44	:	20	21	20	20	41	41	20	44

and so on...

## Error Recovery

1. If FCOMP is unable to locate either of the files, it displays the following message and returns to the operating system.

Error in opening file 'filename'.

2. If either filename is missing, FCOMP displays:

FCOMP requires two parameters (filenames)

# FDUMP File Dump in Hex and ASCII Formats

# FDUMP

## Command Format

FDUMP <filename> {offset}

## Tasks Performed

FDUMP displays the contents of a file in hexadecimal and ASCII formats.

## Using the FDUMP Utility

1. To display the entire file,

TYPE:

FDUMP filename and press RETURN

- filename is an unambiguous file reference.

FDUMP will display the contents of the file in hexadecimal and ASCII formats.

2. To display part of the file,

TYPE:

FDUMP filename offset and press RETURN

- offset is the starting number of the 128 byte record to be displayed. Record 0 is the first record in the file.

The leftmost column indicates the byte count (hexadecimal).

## Example

Dump contents of the file CPMSRN in hexadecimal and ASCII formats, starting at record 17. The console dialogue is as follows:

## Error Recovery

1. If filename is not present in the calling sequence, FDUMP displays:

Expects 'filename' {offset}

2. If the file cannot be found, FDUMP displays:

Unable to open file filename

## FDUMP CPMSRN 17 RETURN

FDUMP v1.1 26 Mar 82

```

                                Record 17
0000 53 43 52 49 50 54 48 4F-4E 02 0D 0A 0D 0A 13 4F  DESCRIPTION....O
0010 56 45 52 56 49 45 57 13-0D 0A 0D 0A 54 68 69 F3  OVERVIEW....This
0020 A0 20 53 6F 66 74 77 61-72 E5 A0 20 52 65 6C 65  . Software. Rele
0030 61 73 E5 20 4E 6F 74 69-63 E5 20 64 65 73 63 72  ase Notice descr
0040 69 62 65 F3 20 56 65 72-73 69 6F EE 20 32 2E 32  ibes Version 2.2
0050 2E B2 A0 20 6F E6 A0 20-43 50 2F 4D AE 20 20 8D  .3. of. CP/M.
0060 0A 56 65 72 73 69 6F EE-A0 20 32 2E 32 2E B2 20  .Version.. 2.2.3
0070 69 F3 20 61 EE 20 75 70-64 61 74 E5 20 6F E6 20  is an update of
```

and so on...

## **FILESET = Create Alphabetical List of Filenames for Use with ARCHIVE**

## **FILESET**

### **Command Format**

FILESET

### **Tasks Performed**

FILESET creates two files containing alphabetized lists of the specified filenames. FLOAD uses one of these files and the FBACK command of ARCHIVE uses the other file.

### **Using the FILESET utility**

TYPE:

FILESET and press RETURN.

FILESET asks for the names of the files to be alphabetized as follows:

Ambiguous filename?

TYPE:

filename and press RETURN

- filename may be ambiguous. An optional disk name may precede the filename.

FILESET asks:

Output filename (extensions will be .FLO and .FBA)

TYPE:

filename and press RETURN.

- filename is the unambiguous primary name of the files to be created: filename.FLO and filename.FBA

FILESET displays the number of filenames matching the ambiguous filename. The following message appears:

Getting matching filenames ... n files match.

- n is the number of files found.

FILESET proceeds to create two files. Each filename entry in the .FBA file is an ARCHIVE command:

FBACK filename

### **Example**

The following is a sample of console dialogue.

Create two files SYSUTIL.FBA and SYSUTIL.FLO containing the names of all system utility programs.

FILESET and press RETURN

Ambiguous filename? \*.COM

Output filename (extensions will be .FLO and .FBA)

**SYSUTIL RETURN**

Getting matching filenames ... 36 files match.

Writing output file **SYSUTIL.FBA**.

Writing output file **SYSUTIL.FLO**.

To backup the files with SYSUTIL.FBA,

TYPE:

ARCHIVE SYSUTIL.FBA

To restore the files,

TYPE:

ARCHIVE SYSUTIL.FLO

Refer to the ARCHIVE utility for instructions on how to answer the questions asked by ARCHIVE.

# FILESET

# FILESET

## Error Recovery

If the files already exist, FILESET asks:

File filename.FBA exists. Append (Y/N)?

TYPE:

Y

and FILESET inserts the filename(s) in the already existing files. FILESET maintains the alphabetical order of the entries.

TYPE:

N

and FILESET asks:

Overwrite (Y/N)?

TYPE:

N

and FILESET returns to the operating system.

TYPE:

Y

and FILESET recreates the files.

If the output filename is not ambiguous, FILESET displays:

Invalid filename.

and repeats the question.

If no filenames match the ambiguous filename, FILESET displays:

Getting matching filenames ... No files match.

and returns to the operating system.

## Points To Remember

The .FBA and .FLO files may be edited using the ED or any other text editor to add, change, or remove entries from it.

FILESET works only with files in the current user directory as does ARCHIVE. To backup the entire disk, move (using the MOVEF utility) all files to one directory and use FILESET in that directory.

## FINDALL Search Disk for File(s)

## FINDALL

### Command Format

FINDALL <filename>

### Tasks Performed

FINDALL searches one or more logical disks for a particular file or set of files.

### Using the FINDALL Utility

To search for a file,

TYPE:

FINDALL filename and press RETURN

- filename may be ambiguous.

When all files are located, a map is displayed showing asterisks in all user directories containing files matching the requested filename.

The user may optionally restrict the search to a single logical disk by including the disk name as part of the filename. In this case, the map (and the search time) is abbreviated.

FINDALL is not affected by the status (public or private) or the files in User 0.

### Example

The following samples of console dialogue show the resulting screen display.

1. Find all .TXT files.

FINDALL \*.TXT RETURN

```
          0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
A:      *           *
B:           *           *
C:                * * *
```

2. Find all \*.LST files on disk B.

FINDALL B:\*.LST RETURN

```
          0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
B:      *           *
```

### Error Recovery

If the file is not found, the message:

File not found.

is displayed.

**Command Format**

FREEZE {SYSTEM}

**Tasks Performed**

After the user has set the system configuration parameters using the SPEED, ASSIGN, OPTIONS and PROTOCOL utilities, FREEZE makes the changes permanent; they become the new defaults when the system is reset or powered-on.

FREEZE does this by copying the operating system to the disk reserved area. Any information kept in memory such as the above parameters and DATE and TIME will be made permanent. Hence do not set the DATE or TIME prior to executing FREEZE unless you desire default date and time to be set upon boot.

**Using the FREEZE Utility**

1. To display information about FREEZE,

TYPE:

FREEZE and press RETURN

FREEZE displays helpful information on the console.

2. To save the current configuration options (See SPEED, ASSIGN, OPTIONS, PROTOCOL);

TYPE:

FREEZE SYSTEM or FREEZE S and press RETURN

FREEZE displays the following messages:

Loading System...

Re-Writing System...

**Error Recovery**

FREEZE terminates with no message if any character other than 's' is entered.

**Command Format**

INVIS <filename> {user}

**Tasks Performed**

INVIS moves all designated files from the current or specified user directory to User 0 and sets their status to SYS and R/O.

**Using the INVIS Utility**

1. To move file(s) to User 0 from the current user directory,

TYPE:

INVIS filename and press RETURN

- filename is a possibly ambiguous file reference.

2. To specify the directory of the file(s) to be moved,

TYPE:

INVIS filename user and press RETURN

- user is the user directory number from which to move the file.

**Example**

Implement the 'public files' on User 0 using FINDALL and INVIS with the following console dialogue.

```
FINDALL *.COM RETURN
```

```

      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
A:    *
B:          *
C:                * * *
```

```
INVIS A: *.COM 10 RETURN
```

```
INVIS B: *.COM 2 RETURN
```

```
INVIS B: *.COM 11 RETURN
```

```
INVIS C: *.COM 4 RETURN
```

```
INVIS C: *.COM 5 RETURN
```

```
INVIS C: *.COM 6 RETURN
```

Files are moved **without** copying to User 0 and their status set to SYS and R/O status. Files in User 0 are now available to all directories.

**Error Recovery**

If no file is specified, INVIS displays:

Proper call is INVIS filename (user).

If no file matching filename can be found, INVIS displays:

File: filename not found on user n.

If an invalid user directory is specified, INVIS displays:

Invalid source number.

**Points To Remember**

INVIS does not move files from one disk to another. Use the PIP utility to transfer .COM files to User 0 on disk A from another disk. Before doing so, make sure that the file names will be unique.

There should be only one user on the system when running INVIS under MP/M.

## **LOAD** Create Executable Program From Assembler Output

## **LOAD**

### **Command Format**

LOAD <filename>

### **Tasks Performed**

Creates an executable program from a file containing 8080 machine language code.

### **Using the LOAD Utility**

To create an executable program,

TYPE:

LOAD filename and press RETURN

- filename is the primary name of a file containing 8080 machine language code. The file is assumed to have a .HEX secondary name. It may be preceded by an optional disk name.

LOAD creates a file named filename .COM which can be executed simply by typing

filename

### **Example**

The following is a sample of command format.

Create an executable program from the file TEST.HEX.

LOAD TEST

Execute the program.

TEST

### **Error Recovery**

If the .HEX file cannot be found, LOAD displays:

ERROR: CANNOT OPEN SOURCE, LOAD ADDRESS n

### **Points To Remember**

The contents of the .HEX file must be valid Intel format hexadecimal machine code records which begin at address 100H. The addresses must be in ascending order. Gaps in addresses are filled in with zeroes by the LOAD Utility.

## **LOADREL Load Software Release Tape from Onyx**

## **LOADREL**

### **Command Format**

LOADREL

### **Tasks Performed**

LOADREL loads software files from an Application or Language Product Tape onto disk A on a file by file basis.

### **Using the LOADREL Utility**

TYPE:

LOADREL and press RETURN

If the tape has not been inserted in the tape drive, LOADREL displays:

Load tape and type carriage return.

LOADREL then prompts:

How many tensioning passes required?

**NOTE:** Usually three tensioning passes are required when using a tape for the first time. At other times, no tensioning is necessary unless a large number of tape errors were previously encountered.

PRESS:

RETURN

if no tensioning is required.

TYPE:

n and press RETURN

- n is the number of tensioning passes.

if tensioning is required.

After performing any requested tensioning, LOADREL begins loading the files from the tape. For each file loaded, LOADREL displays:

Loading filename

with the appropriate file name. If the file that LOADREL is attempting to load already exists on the disk, LOADREL prompts:

File filename exists on disk. Overwrite?

TYPE:

Y

and LOADREL erases the existing file and loads the file. The previous file is lost. For any other response, LOADREL prompts:

New name for file (CR = do not load file)?

PRESS:

RETURN

and LOADREL will not attempt to load the named file, but displays:

File not loaded.

TYPE:

a valid filename and press RETURN

and LOADREL attempts to load the file using that name. If this file also exists, LOADREL again asks if the user wants to overwrite the file.

The above dialogue is repeated for each file on the tape. At the end of the load, LOADREL displays:

Software Load Complete.

and returns to the operating system.

## LOADREL

## LOADREL

### Example

Load file ACCTXY3.GSM from an Application Product tape. The console dialogue is as follows.

```
Loarel - Loads Release Tape. 1.3 26 Mar 82
Copyright ©Johnson-Laird, Inc.
How many tensioning passes required (CR = 0)? RETURN
Loading ACCTXY3.GSM
Software Load complete.
```

(Tape Statistics are provided to monitor quality)

Error Type	( Recovery =>	Easy	Hard )
CRC Errors		0000	0000
Failed to detect data		0000	0000
Data but no synch char.		0000	0000
Waited too long for char.		0000	0000
Waited too long for CRC		0000	0000
Read-after-write data err.		0000	0000

### Error Recovery

If LOADREL cannot read the tape header it displays:

```
Error -
Not a valid Software Release Tape
```

and returns to the operating system.

Refer to Section 3.4 on Hardware Errors for explanations of any disk or tape errors which may occur.

### Points To Remember

LOADREL expects a 'serialized' tape from Onyx. Use ARCHIVE to transfer files between systems.

## **MOVEF** Move File(s) between Users

## **MOVEF**

### **Command Format**

MOVEF <filename> <destination> {source}

### **Tasks Performed**

Moves file(s) from one user directory to another without copying them.

### **Using the MOVEF Utility**

1. To move files from the current directory,

TYPE:

MOVEF filename destination and press RETURN

- filename may be ambiguous, in which case all files matching the filename will be moved. An optional disk name may precede the filename.
- destination is the user directory number to which the file(s) will be moved.

MOVEF searches the current user directory for the named file(s) and changes their directory entries to the destination directory number.

2. To move files from any directory,

TYPE:

MOVEF filename destination source  
and press RETURN

- source represents the user directory number **from** which the file(s) will be moved.

### **Examples**

The following are samples of command formats. Each is entered followed by a RETURN.

1. Transfer all .LST files to User 10 from the current directory.

```
MOVEF *.LST 10
```

2. Transfer all .COM files to User 0 from User 2.

```
MOVEF *.COM 0 2
```

MOVEF can be used in conjunction with FINDALL to transfer files from several directories. See the example with the FINDALL utility.

### **Error Recovery**

1. If the filename or destination directory number is missing, MOVEF displays:

```
No file specified
```

or

```
Destination user not specified
```

along with the proper calling sequence.

2. If the source and destination directories are the same, MOVEF displays:

```
Source and destination are the same
```

3. If the file cannot be found, MOVEF displays:

```
File filename not found on user n
```

4. If an illegal value for either user directory is specified, MOVEF displays:

```
Invalid source user
```

or

```
Invalid destination user
```

## **MOVEF**

### **Points To Remember**

MOVEF moves files between user directories on the same logical disk. It is necessary to use the PIP utility to transfer files from one disk to another.

Using MOVEF to transfer files between directories can cause a problem with other utilities since it is possible that two files with the same name will be listed in a directory. It is important to determine the uniqueness of a filename with FINDALL before using MOVEF.

There should be only one user on the system when running MOVEF under MP/M.

## **MOVEF**

**Command Format**

OPTIONS

**Tasks Performed**

To enable or disable display of the current user number each time CP/M is warm-booted (each time CP/M is reloaded into memory after a program terminates) and accepts (in hexadecimal format) the special characters used by the DO command to represent carriage returns (RETURN) and CTRL + C. The defaults are '\'  
(Hex 5C) and '\'  
(Hex 7C) respectively.

**Using the OPTIONS Utility**

TYPE:

OPTIONS and press RETURN

OPTIONS displays:

This is the systems options utility

Enter y to change the Current User Display on Warm Boot:

TYPE:

N

for no change to the current user display.

TYPE:

Y

and OPTIONS prompts:

Enter y if the Current User No. is to be displayed after a warm boot.

TYPE:

N

for no display.

TYPE:

Y

for a display after each warm boot.

OPTIONS next prompts:

Enter y to change Control Characters used with DO command:

TYPE:

N

for no change to the control characters.

TYPE:

Y

for a control character change, and OPTIONS prompts:

Enter (in hex) character to be translated into Carriage Return:

and

Enter (in hex) character to be translated into Control-C:

In both cases the response must be the **hexadecimal equivalent** of the ASCII character to be used followed by RETURN (see Example 2).

**Examples**

The following are samples of console dialogue.

1. Disable the current user message.

**OPTIONS RETURN**

This is the systems options utility

Enter y to change to Current User Display on Warm Boot: **Y**

Enter y if the Current User No. is to be displayed after a warm boot: **N**

Enter y to change control characters used with DO command: **N**

2. Change characters used with the DO utility and enable current user display.

**OPTIONS RETURN**

This is the system options utility

Enter y to change the Current User Display on Warm Boot: **Y**

Enter y if the Current User No. is to be displayed after a warm boot: **Y**

Enter y to change Control Characters used with DO command: **Y**

Enter (in hex) character to be translated into Carriage Return: **5C RETURN**

Enter (in hex) character to be translated into Control-C: **7C RETURN**

**Error Recovery**

1. OPTIONS ignores any character but 'Y' in response to the messages:

Enter y to change the Current User Display or Warm Boot:

Enter y if the Current User Number is to be displayed after a warm boot.

Enter y to change Control Characters used with DO command:

2. OPTIONS accepts any character(s) in response to the messages:

Enter (in hex) character to be translated into Carriage Return:

Enter (in hex) character to be translated into Control-C:

No error message is given if the character is not a valid hex representation.

**Points To Remember**

The options selected remain in force only until the next system reset or cold start. The selected options can be made permanent by using the FREEZE utility. This removes the need to rerun the OPTIONS utility.

**Command Format**

DDT PATCH.COM

**Tasks Performed**

PATCH allows installation of corrections to the operating system as documented in a Software Problem Report.

Use of this utility requires some knowledge of the standard CP/M utility DDT (Dynamic Debugging Tool).

**Using the PATCH Utility**

The patching operation is performed by typing in the appropriate dialogue exactly as described in the Product Support Note, as Table 3-4 illustrates in the following example.

**Example**

It is necessary to change the contents of a specific location in CP/M to a value of 00. The location and the current contents are identified in a Product Support Note sent to all registered users, as are exact instructions and console dialogue.

Table 3-4. Patch Dialogue Example

Dialogue	Function
DDT PATCH.COM	Load the PATCH utility
DDT VERS 2.0	
NEXT PC	
0280 0100	DDT executes the load
-G RETURN	Start execution of PATCH
System Patch Utility Vn 1.1	28 Mar 82
CCP is loaded at 3000H	
BIOS is located at 4600H	
Relocation bit map is located at 5388H	
Make patches and continue with the "G" command	
*010E	Control returns to DDT
-S4703 RETURN	Address specified in Software Problem Report
4703 31 00 RETURN	Address followed by - Old contents - followed by user entered new contents
4704 22 . RETURN	.terminates S command
-G RETURN	Restart PATCH again
Patched system being written to reserved track	

**Error Recovery**

See the CP/M DDT User's Guide for DDT command errors.

## Command Format

PIP {command}

## Tasks Performed

Copies, displays, prints, and combines disk files.

## Using the PIP Utility

1. To perform more than one operation,

TYPE:

PIP and press RETURN

PIP reads commands directly from the console, prompted with the '\*' character. Each successive command causes some operation to take place as indicated in the PIP Guidelines that follow.

To terminate PIP,

PRESS:

RETURN

2. To perform a single operation,

TYPE:

PIP command and press RETURN

PIP terminates immediately after processing the command with no further prompting commands.

3. The form of each command is:

destination = source#1 {parameters}, source#2 {parameters},..., source#n {parameters}

- destination is the file, or name of the physical or logical device that is to receive the data.
- source#n represents a series of one or more files or devices which are copied (beginning with the leftmost) to the destination.
- parameters are one or more designators enclosed in '()' which control the operation.

## 4. Abbreviated Commands

PIP allows abbreviated commands for transferring files between logical disks. Table 3-5 shows the abbreviated form of each command, and how it functions.

**Table 3-5. Abbreviated PIP Commands**

Command Form	Function
x:=filename	Copies all files matching the possibly ambiguous filename from the current default disk to disk x (A, B...).
x:y:filename	Copies all files matching the possibly ambiguous filename from disk y to disk x.
filename=y:	Copies all files matching the possibly ambiguous filename from disk y to the current default disk.
x:filename=y:	Copies all files matching the possibly ambiguous filename from disk y to disk x when y explicitly identifies the source disk.
NOTE:	If the filename is ambiguous, PIP lists each filename which satisfies the reference as it is being copied. Any existing file with the same name as the destination file is replaced by the copied file.

**PIP Guidelines**

1. When there are multiple source files in the command line, the individual files are assumed to contain ASCII characters, with an assumed operating system end-of-file character (CTRL + Z) at the end of each file (see the O parameter to override this assumption).
2. Lower case alphabetic characters in the command are translated to upper case.
3. Total command line length must not exceed 255 characters; using a CTRL + E forces a physical carriage return for lines exceeding the console width.
4. PIP accepts, for destination and sources, file names (with or without a preceding disk name), physical device names or logical device names. Refer to Section 3.2.
5. Multiple source files are concatenated from left to right.
6. A destination file entry can appear as one or more of the source files. In this case, PIP does not alter the source file until all concatenation has taken place.
7. Any existing destination file is removed if the command is entered correctly. It is not removed if an error condition arises.
8. Parameters are enclosed in square brackets, '()', and must occur immediately after the affected file or device name. Except for the S and Q parameters, PIP allows an optional decimal integer following each parameter. Table 3-6 lists all valid PIP parameters, and describes their corresponding action.

Table 3-6. PIP Parameters

Parameter	Function	Parameter	Function
B	Block mode transfer: Holds the data until an ASCII X-OFF character (CTRL + S) is received from the source device allowing transfer of data from a continuous reading device to a disk file. An error message occurs if the buffers overflow.	N	Adds line numbers to each line transferred to the destination starting at one (1), and incrementing by one (1). The number is displayed without leading zeros, and is followed by a ':'. If N2 is specified, then leading zeros are included, and a tab is inserted following the number. Using the T parameter expands the tabs.
Dn	Deletes the characters extending beyond the column specified by 'n' during a data transfer operation.	O	Object file (non-ASCII) transfer: Ignores the normal operating system end of file character.
E	Echo: monitors all transfer operations on the console as they are executing.	Pn	Inserts page ejects for every n lines after an initial page eject. If n = 1 or is excluded altogether, page ejects occur every 60 lines. Pn ignores any formfeeds present in the file.
F	Filter formfeeds from the file: Removes all imbedded formfeeds; allow new formfeeds to be inserted simultaneously with the P parameter.	Qsiz	Quits copying from the source device or file when it encounters the string 's' (terminated by CTRL + Z).
Gn	Retrieves the file(s) from the user directory specified by n.	R	Reads system files.
H	Hexadecimal data transfer: Checks all data for correct Intel hexadecimal file format, removes any non-essential characters between hexadecimal records during the copy operation, and prompts for guidance from the console for corrective action if errors occur.		
I	Ignore: Bypasses '.00' records during an Intel hexadecimal format file transfer. The I parameter automatically sets the H parameter.		
L	Lower case: Translates upper case alphabetic characters to lower case.		

---

<b>Parameter</b>	<b>Function</b>
<b>Ssiz</b>	<p>Starts copying from the source device when it encounters the string 's' (terminated by CTRL + Z). A portion of the selected file (such as a subroutine), can be extracted by using the S and Q parameters. The start and quit strings are always included in the copy operation.</p> <p>NOTE: The strings following the s and q parameters are translated to upper case if form (2) of the PIP command is used. Form (1) does not perform this upper case translation automatically.</p> <p>(1) PIP (2) PIP &lt;command line&gt;</p>
<b>Tn</b>	<p>Expands tabs (CTRL + I characters) to every nth column during the transfer of characters from source to destination.</p>
<b>U</b>	<p>Upper case characters: Translates lower case alphabetic characters to upper case during the copy operation.</p>
<b>V</b>	<p>Rereads data following the write operation to verify the copy. The destination must be a disk file.</p>
<b>W</b>	<p>Writes over R/O files without prompting the user. If there are several source files it is necessary to include the W parameter only with the last file in the list.</p>
<b>Z</b>	<p>Zero the 8th bit (parity) on input for each ASCII character.</p>

---

## PIP

### Examples

The following are samples of command formats. Each is entered followed by a RETURN.

1. Copy file Y to file X; Y remains unchanged, both on the default disk.

```
X=Y
```

2. Concatenate files Y and Z and copy to file X, with Y and Z unchanged. All files are on the current default disk.

```
X=Y, Z
```

3. Copy OLD.DAT from disk B to the current disk and change the name to NEW.DAT.

```
NEW.DAT=B:OLD.DAT
```

4. Copy and verify all files with the extension .COM from the current disk to disk B.

```
B:=*.COM(V)
```

5. Copy the file TEST.ASM to disk A from disk B when the default disk is A.

```
TEST.ASM=B:
```

```
(Equivalent to A:TEST.ASM=B:TEST.ASM)
```

6. Copy the file MENU.BAS from disk A to disk B when the default disk is A.

```
B:=MENU.BAS
```

```
(Same as B:MENU.BAS=A:MENU.BAS)
```

7. Display X.PRN on the console device

```
CRT:=X.PRN
```

8. Concatenate three BASIC files and copy to the console.

```
CON:=X.BAS,Y.BAS,Z.BAS
```

## PIP

9. Print X.ASM, on the printer: number each line, expand tabs to every eighth column, and translate lower case alphabetic characters to upper case.

```
LST:=X.ASM(nT8U)
```

An equivalent statement is PRN:=X.ASM

### Error Recovery

Depressing any key on the keyboard stops the copy operation at any point. The message, ABORTED, appears, indicating an incomplete operation.

When an operation is terminated (or if an error occurs), commands set up with the SUBMIT command are removed by PIP.

### Points to Remember

Only the abbreviated commands allow the use of ambiguous filenames.

## **PROTOCOL** Set/Display Serial Port Protocol

## **PROTOCOL**

### **Command Formats**

```
PROTOCOL  
PROTOCOL <physical-port> {protocol-type}
```

### **Tasks Performed**

Changes the operating system's method of communicating with a serial device attached to one of the serial ports. This prevents the CPU from sending data to a peripheral before the peripheral is ready to receive it.

### **Description of Protocols**

The operating system, over and above the protocol set with this utility, responds to the XON/XOFF protocol. This is imposed on all output that is initiated by it.

#### **XON/XOFF Protocol**

Before each character is output from the computer to the peripheral device, the software checks to see whether there is any incoming data from the peripheral. If the incoming character is XOFF, no further output will be made to the peripheral until a subsequent XON character is input.

#### **DTR Protocol**

The software also checks for Data Terminal Ready (DTR) before each character output from the computer to the peripheral device. If pin 20 on the 25-pin computer-peripheral connector is positive or at a high voltage, the data character is sent.

This prevents the CPU from sending output while the voltage is low (peripheral not ready).

### **Using the PROTOCOL Utility**

1. To display the options available with PROTOCOL, TYPE:

```
PROTOCOL and press RETURN
```

Descriptions of all the possible parameters that can be used with PROTOCOL are then displayed on the console.

2. To establish a protocol between the CPU and a serial port,

TYPE:

```
PROTOCOL physical-port protocol-type  
and press RETURN
```

- physical-port is one of the following:
  - CONSOLE or TERMINAL (the terminal port)
  - MODEM (the modem port)
  - SERIAL (the serial printer port)
- protocol-type can be one of the following:
  - NONE (only the XON/XOFF in force)
  - DTR
  - XON

## PROTOCOL

Only the first character of either of the parameters (in upper or lower case) need be entered.

3. To display the current protocol assignment of a physical port,

TYPE:

PROTOCOL physical-port and press RETURN

PROTOCOL displays the current protocol in force for the specified physical port.

### Examples

The following are samples of command formats. Each is entered followed by a RETURN.

1. Enable output to the console port only when DTR (pin 20) is high.

PROTOCOL CONSOLE DTR or protocol c d

2. Check the current protocol in use with the modem port.

PROTOCOL MODEM or protocol m

### Error Recovery

1. PROTOCOL displays:

X is not a valid physical name

and lists the correct alternatives if the physical port name is incorrect.

2. PROTOCOL displays:

Z is an illegal protocol

and lists the correct alternatives if the protocol name is incorrect.

## PROTOCOL

### Points To Remember

The protocols set using this utility go into effect immediately, but remain in effect only until the next system reset.

PROTOCOL is used only with serial ports.

The FREEZE utility makes protocol selection permanent. This removes the need to set protocols for the serial devices after each system reset or power-on.

MP/M does not support multiple terminal types, therefore, PROTOCOL applies to all terminals on system.

**Command Formats**

PUB  
PRIV

**Tasks Performed**

To enable and disable the User 0 public files facility. Public files in User 0 are accessible to every other user as though they resided in that user's own space.

**Using the PUB/PRIV Utilities**

1. To enable files in User 0 to be accessible from any user directory.

TYPE:

PUB and press RETURN

The files on disk A in User 0 will now be 'seen' by the DIR command in any user directory on disk A. Similarly the files on disk B on User 0 will be 'seen' by the DIR command in any user directory on disk B. The files are not altered in any way by these utilities.

2. To disable public file access,

TYPE:

PRIV and press RETURN

Each user space is now logically independent of all others.

**Error Recovery**

None needed.

**Points To Remember**

**WARNING:** When PUB is in effect the operating system does not distinguish between files in User 0 and the current user directory. Thus, if there were a file named INFO.DAT in the current user directory and in User 0 it would appear twice in the directory listing after using PUB. If that file were then ERASed, the operating system would remove both copies. For this reason, files should be renamed or rearranged with PIP before invoking PUB. Similarly, other utilities such as ARCHIVE do not execute properly when duplicate filenames occur in a directory while PUB is in effect.

It is recommended that all frequently accessed program and data files be put in User 0 and set to system (SYS) and read-only (R/O) status. See the INVIS utility. Files set to SYS status will not appear in any DIR listing (even User 0). Files set to R/O status cannot be inadvertently ERASed or overwritten.

PUB and PRIV affect all logical disks. Making files 'public' in User 0 on disk A will make files in User 0 on disk B 'public' also.

If PRIV is currently in effect, PUB can only be initiated from User 0 (unless a copy is kept in another directory).

### Command Format

QERA <filename>

### Tasks Performed

Erases files. Unlike the ERA command, this utility displays each file name before erasure and prompts the user to erase or retain the file.

### Using the QERA Utility

TYPE:

QERA filename and press RETURN

- filename is a possibly ambiguous file reference preceded by an optional disk name.

QERA scans the current directory and, for each file encountered that matches the filename, displays:

Erase filename?

TYPE:

Y

to erase the file, and QERA displays:

<===Erased

to the right of the response.

TYPE:

N

and QERA displays:

<===Not erased

and proceeds to the next file.

### Example

The following samples of console dialogue show the resulting screen displays.

1. Erase all .BAK files.

**QERA \*.BAK RETURN**

QERA v1.1 1v1.11

Erase file DOCDIST .BAK ? Y <=== Erased.

2. Selectively erase all .BAK files.

**QERA \*.BAK RETURN**

QERA v1.1 1v1.11

Erase file QUAL .BAK ? N <=== Not Erased!

Erase file COMPET .BAK ? Y <=== Erased

Erase file CMPV2 .BAK ? N <=== Not Erased!

### Error Recovery

If the requested file is not found, QERA displays:

File not found.

and returns to the operating system.

If the requested filename is missing, QERA displays:

Filename required.

## **REN** Rename File

## **REN**

### **Command Format**

```
REN <new-filename> = <old-filename>
```

### **Tasks Performed**

Change the name of a file.

### **Using the REN Command**

TYPE:

```
REN new-filename=old-filename  
and press RETURN
```

- new-filename is the unambiguous **new** name for the file.
- old-filename is the unambiguous **current** name for the file.

The file 'old-filename' is changed to 'new-filename'.  
An optional disk name may precede either or both filenames.

### **Examples**

The following are sample command formats. Each is entered followed by a RETURN.

1. Rename file Q.R to X.Y.

```
REN X.Y = Q.R
```

2. Rename file Y.ASM to X.ASM on disk A.

```
REN A:X.ASM = Y.ASM
```

3. Rename the file A.BAK to A.ASM on disk B.

```
REN B:A.ASM = B:A.BAK
```

### **Error Recovery**

If the file 'new-filename' already exists, the REN command displays:

```
FILE EXISTS
```

and does not perform the change.

If the file 'old-filename' does not exist in the current directory, the REN command displays:

```
NOT FOUND
```

### **Points To Remember**

If either filename is preceded by a disk name, the other file is assumed to exist on the same disk. If both filenames are preceded by disk names, the same disk must be specified.

REN works only on single (unambiguous) files.

**Command Format**

SAVE <n> <filename>

**Tasks Performed**

Copies the current version of the program in memory onto the disk.

**Using the SAVE Command**

To create a file containing the version of the program currently in memory,

TYPE:

SAVE n filename            and press RETURN

- n is the size of the program in 256 byte pages (decimal). The number of pages to be copied is determined from the leading hexadecimal digit(s) of the highest program address.
- filename is an unambiguous file reference which may be preceded by an optional disk name.

SAVE creates a file and copies the contents of the user memory area into it, allowing the program to be executed at a later time.

**Example**

The following are sample tasks showing the command formats.

1. Save the contents of memory in the file TEST.COM. The highest address of TEST.COM is 37AC. The number of pages is 37H which is  $3 * 16 + 7 = 55$  decimal pages.

SAVE 55 TEST.COM

2. Save the contents of memory in the file PROG.COM on disk B. The highest address of PROG.COM is 8AC. The number of pages is 8 which is  $0 * 16 + 8 = 8$  decimal pages.

SAVE 8 B:PROG.COM

**Error Recovery**

If the file already exists, SAVE overwrites it. No warning message is displayed. If n is not specified SAVE displays:  
filename?

No error message is displayed if the filename is omitted or if SAVE is called without any parameters.

## Command Format

SPARE

## Tasks Performed

Displays the current contents of the spare table and, if desired, adds a specific physical sector (512 bytes) to the spare table. If a sector gives CRC errors when reading, you will need to add another physical sector.

## Using the Spare Utility

TYPE:

SPARE and press RETURN

If the system has more than one physical disk drive, SPARE requests the physical disk number of the spare table to be manipulated. The first disk is number 0.

If the SPARE table is full, SPARE displays:

SPARE DIRECTORY FULL

If there are less than eight available entries remaining, SPARE displays:

SPARE DIRECTORY ALMOST FULL

SPARE displays the current spare table and then displays:

Enter Y if you wish to add spare sectors to the directory

TYPE:

N

and SPARE returns to the operating system.

TYPE:

Y

and SPARE displays:

Enter logical sector (hex) e.g. ss ss ss

TYPE:

ss ss ss and press RETURN

- ss ss ss is the track number from the DISK ERROR MESSAGE with leading zeros added (if necessary) to make a 6 digit hexadecimal number. Place a space between each pair of hexadecimal digits.

SPARE displays:

About to Spare Disk n Logical Sector sssss  
Enter Y to spare this sector:

TYPE:

Y

to add the sector to the spare table and SPARE displays:

Press RETURN to see new spare table:

PRESS:

RETURN

and SPARE displays the new spare table and the message:

Requested sector has now been spared  
RE-BOOT SYSTEM TO USE NEW SPARE TABLE

## Example

The following example of console dialogue is a request to display the SPARE table and add a sector to it. The disk error message was:

DISK ERROR = (CODE, DISK, TRK, SEC)  
04 0A 02E7 01

(04 = CRC error code, 0A = disk, 02E7 = 512-byte physical sector, 01 = 128-byte logical sector)

## SPARE

### SPARE RETURN

Spare Table:

base address of alternate spare sectors: 000001

Sectors spared: 1

Unused spare sectors: 30

(Bad cylinder: bad head/sector:alternate head/sector)

	First	Second	Third	Fourth
0:	008E:71:00	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
4:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
8:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
12:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
16:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
24:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
28:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF

Enter y if you wish to add spare sectors to the directory.

Y

Enter logical sector (hex) e.g. ss ss ss **00 02 E7 RETURN**

About to Spare Disk 0 Logical Sector 0002E7

Enter Y to spare this sector: Y

Hit RETURN to see new Spare table: RETURN

Base address of alternate spare sectors: 000001

Sectors spared: 2

Unused spare sectors: 29

(Bad cylinder:bad head/sector:alternate head/sector)

	First	Second	Third	Fourth
0:	008E:71:00	0030:60:00	FFFF:FF:FF	FFFF:FF:FF
4:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
8:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
12:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
16:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
24:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF
28:	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF	FFFF:FF:FF

Requested sector has now been spared  
RE-BOOT SYSTEM TO USE NEW SPARE TABLE

## SPARE

### Error Recovery

If the sector number is invalid or the hardware is malfunctioning, SPARE displays:

Can't read that sector due to error other than CRC error

Be sure that you gave the correct sector number.

DID NOT SPARE THAT SECTOR

If the error is not a CRC error and you have entered the correct sector number, the hardware requires corrective maintenance.

If the sector can be read successfully, SPARE displays:

I can read that sector ok!

Are you SURE you want to spare it?

About to spare Disk n Logical Sector ssssss

Enter Y to spare this sector:

If the sector is not to be added to the spare table,

TYPE:

N

and SPARE displays:

SECTOR NOT SPARED

and asks again for a sector number. To exit SPARE at this point,

PRESS:

CTRL + C

### Points To Remember

NOTE: After a bad sector has been included in the spare table the system must be reset (using the reset button) in order for the new table to take effect. The operating system has to be reloaded to force it to read the new SPARE table that has just been created.

**Command Format**

```
SPEED  
SPEED <physical-port> <baud-rate>
```

**Tasks Performed**

To change the baud rates (transfer rates) of the serial ports and the Real-Time Clock 'tick' rate.

SPEED also displays the current baud rate for the serial ports or the tick rate for the clock.

**Using the SPEED Utility**

1. Display the parameter options.

TYPE:

```
SPEED and press RETURN
```

SPEED displays all the parameters.

2. Set the baud rate of a port.

TYPE:

```
SPEED physical-port baud-rate and press RETURN
```

- physical-port is one of the following:
  - CONSOLE or TERMINAL (the terminal port)
  - MODEM (the modem port)
  - SERIAL (the serial printer)
  - RTC (the Real Time Clock)

Only the first character of the physical port name (in upper or lower case) is significant.

- baud-rate is one of the following:

- 19200	or	19
- 9600	or	9
- 4800	or	4
- 2400	or	2
- 1200	or	12
- 600	or	6
- 300	or	3
- ?		Causes current rate to be displayed

The baud-rate may be abbreviated as shown. More than one port may be specified in the command (see Example 3). For the real-time clock, the tick rate is a number in the range 1 to 60. It represents the number of 60th's of a second between each clock tick (and internal counter update). The normal value is 60, i.e., the clock ticks once a second.

The real-time clock can be used as an interval timer by increasing the frequency of the tick rate, i.e., using a number less than 60. For example, by setting the count to 1, the real-time clock will tick every 1/60th of a second. See the description of the TIME utility for more information on how to use the clock as an interval timer.

## SPEED

### Examples

The following are samples of command formats. Each is entered followed by a RETURN.

1. Set the baud rate of the modem port to 9600 (960 characters per second).

```
SPEED MODEM 9600 or speed m 9
```

2. Display the current baud rate setting for the serial printer port.

```
SPEED SERIAL ? or speed s ?
```

3. Set the modem port to 19200 baud, the real-time clock tick rate to once every 1/30th of a second and display the current console baud rate.

```
speed m 19 r 2 c ?
```

or

```
SPEED MODEM 19200 RTC 2 CONSOLE ?
```

## SPEED

### Error Recovery

1. SPEED displays:

Illegal number of parameters (they must be specified in pairs)

if one of the parameters is missing.

2. SPEED displays:

X is not a valid device name

and lists the correct alternatives if the physical port name is incorrect.

3. SPEED displays:

Y is an illegal baud rate

and lists the correct alternatives if an invalid baud rate has been selected.

### Points To Remember

The changes made by SPEED take effect immediately but remain in force only until the system is reset or powered up.

The FREEZE utility makes the changes permanent, and eliminates the need to use SPEED after each system reset.

MP/M does not support terminal types, therefore SPEED applies to all terminals on system.

**Command Format**

STAT  
STAT <command>

**Tasks Performed**

Provides general statistical information about disk and file storage.

Using different forms of STAT also allows file status to be examined and/or altered as well.

**Using the STAT Utility**

1. To display the possible STAT commands,

TYPE:

STAT VAL: and press RETURN

STAT displays:

Temp R/O Disk: d:=R/O  
Set Indicator: d:filename.typ \$R/O \$R/W \$SYS \$DIR  
Disk Status: DSK: D:DSK:  
User Status: USR:  
Iobyte Assign:  
CON: = TTY: CRT: BAT: UC1:  
RDR: = TTY: PTR: UR1: UR2:  
PUN: = TTY: PTP: UP1: UP2:  
LST: = TTY: CRT: LPT: UL1:

The logical device shown to the left can take any of the four physical assignments shown to the right on each line.

**NOTE:** IOByte Device assignments are not implemented in the Onyx version of the operating system. See Section 3.2 for a list of valid logical and physical device names.

2. To display the amount of available storage,

TYPE:

STAT and press RETURN

STAT displays the storage remaining on all currently active disks.

X: R/W, SPACE: nnnK

or

X: R/O, SPACE: nnnK

- X is the disk name.
- R/W indicates the disk may be read or written.
- R/O indicates the disk is read only (a disk becomes R/O by explicitly setting it to read only).
- nnn indicates the space remaining on disk X (given in kilobytes).

3. To display the storage remaining on a disk,

TYPE:

STAT X: and press RETURN

- X is the name of the disk.

This command selects the disk before the storage is computed, allowing the status of any disk to be determined without changing the current default disk. STAT displays:

BYTES REMAINING ON X:nnnK

4. To display the characteristics of all active disks,

TYPE:

STAT DSK:            and press RETURN

STAT displays for each active disk:

X: Drive Characteristics  
 n1: 128 Byte Record Capacity  
 n2: Kilobyte Drive Capacity  
 n3: 32 Byte Directory Entries  
 n4: Checked Directory Entries  
 n5: Records/Extent  
 n6: Records/Block  
 n7: Sectors/Track  
 n8: Reserved Tracks

- X: is the disk name.
- n1 is the total record capacity.
- n2 is the total capacity in kilobytes.
- n3 is the directory size (i.e., the number of files that can be stored on the disk).
- n4 is zero for the hard (Winchester) disk.
- n5 is the number of records per extent. Each file can be no larger than 8 mbytes.
- n6 represents the number of records per block.
- n7 is the number of sectors per track.
- n8 is the number of reserved tracks. Since logical disks can share the same physical disk drive, this number may be quite large as this mechanism is used to skip lower-numbered disk areas allocated to the other logical disks.

5. To display the characteristics of a particular disk,

TYPE:

STAT X:DSK:            and press RETURN

- X is the disk name.

STAT displays the disk characteristics only for disk X.

6. To set a disk to read-only status,

TYPE:

STAT X:=R/O            and press RETURN

- X is the disk name

The R/O status remains in effect until the next warm or cold start takes place.

7. To display the status of a set of files,

TYPE:

STAT filename            and press RETURN

- filename is a possibly ambiguous filename.

When a disk is given ahead of the file name, the specified disk is selected before the directory is searched.

The file(s) which satisfy the filename are listed in alphabetical order, with storage requirements for each file under the heading:

Recs	Bytes	Ext	Acc
rrrr	bbb	ee	ZZZ X:pppppppp.sss

- rrrr is the number of 128-byte records allocated to the file.
- bbb is the number of kilobytes allocated to the file (bbb=rrrr\*128/1024).
- ee is the number of 16K extensions (ee=bbb/16).
- ZZZ is either R/W (Read/Write) or R/O (Read Only) file status.
- X is the disk name containing the file (A, B, etc.).
- pppppppp is the (up to) eight-character primary file name.
- sss is the (up to) three-character secondary name.

If the file status is 'SYS', the filename is surrounded by '( )'. Available storage is shown after the listing of individual files.

8. To set the status of a file,

TYPE:

STAT filename \$XXX and press RETURN

- filename is a possibly ambiguous filename preceded by an optional disk name.
- XXX is one of the following:
  - R/O to set read-only status
  - R/W to set read-write status
  - SYS to set system status
  - DIR to set directory status

9. To display current logical to physical device mapping,

TYPE:

STAT DEV: and press RETURN

which produces a listing of each logical device to the left, and the current corresponding physical device to the right.

10. To change the current logical to physical device assignments,

TYPE:

STAT ld1=pd1, ld2=pd2,..., ldn=pdn  
and press RETURN

- ld1 through ldn are logical device names.
- pd1 through pdn are compatible physical device names. (Refer to the 'VAL' command.)

NOTE: This command has no effect in the Onyx version of the operating system.

11. To display the current user directory number,

TYPE:

STAT USR: and press RETURN

STAT displays:

Active User:n

Active Files:n1, n2,....

- n is the current user number.
- n1,n2...are the user directory numbers containing files on the current default disk.

### Examples

The following samples of console dialogue show the resulting screen display.

1. Display disk usage when only disk A is active.

**STAT RETURN**

A: R/W, Space: 1106k

2. Display disk characteristics when disks A and B are active.

**STAT DSK: RETURN**

A: Drive Characteristics  
 18720: 128 Byte Record Capacity  
 2340: Kilobyte Drive Capacity  
 256: 32 Byte Directory Entries  
 0: Checked Directory Entries  
 128: Records/Extent  
 16: Records/Block  
 8: Sectors/Track  
 150: Reserved Tracks

B: Drive Characteristics  
 26160: 128 Byte Record Capacity  
 3270: Kilobyte Drive Capacity  
 1024: 32 Byte Directory Entries  
 0: Checked Directory Entries  
 128: Records/Extent  
 16: Records/Block  
 8: Sectors/Track  
 2490: Reserved Tracks

3. Display current user directory and directories with files on the current disk.

**STAT USR: RETURN**

Active User: 0  
 Active Files: 0 1 7

4. Display the status of a file, change the status and then display the status again.

**STAT TIME.COM RETURN**

Recs	Bytes	Ext	Acc
50	8k	1	R/W A:TIME.COM

Bytes Remaining on A: 1106k

**STAT TIME.COM \$SYS RETURN**

**STAT TIME.COM RETURN**

Recs	Bytes	Ext	Acc
50	8k	1	R/W A: (TIME.COM)

**Error Recovery**

1. If the disk is not present on the system, STAT displays:

BDOS Err on X: Select

PRESS:

RETURN

to return to the operating system.

2. If STAT cannot find a requested file, it displays:

FILE NOT FOUND

**Points To Remember**

When a disk has been set to read-only by STAT, the message:

BDOS Err on X: READ ONLY

appears if there is an attempt to write to the read-only disk X.

When a file is read-only, attempts to erase or alter it cause the message:

BDOS Err on X: File R/O

# SUBMIT Command File Processor

# SUBMIT

## Command Format

```
SUBMIT <filename> {parameter 1} {parameter 2} ...{parameter n}
```

## Tasks Performed

Allows operating system commands to be batched together for automatic processing.

## Using the SUBMIT Utility

TYPE:

```
SUBMIT filename parameter 1 parameter 2...  
parameter n and press RETURN
```

- filename is the primary name of a file with a secondary file name of .SUB.
- parameter n is the actual parameter to be used wherever \$n occurs in the .SUB file.

The .SUB file contains operating system commands, with interspersed '\$' parameters of the form

```
$1 $2 $3 ... $n
```

corresponding to the actual parameters to be included when the file is submitted for execution.

The SUBMIT function creates a file of substituted commands with the name

```
$$$SUB
```

on disk A. Once SUBMIT has terminated, the \$\$\$SUB command file becomes the source of input, rather than the console, to the operating system. The last command in a .SUB file may be the SUBMIT command, allowing another .SUB file to be processed.

**NOTE:** Dollar signs may be introduced into a SUBMIT file by using the ED (or any other editor) utility, and typing a \$\$ (this reduces to a single \$ within the command file). Further, an up-arrow symbol '^' may precede an alphabetic character 'X', to produce a single CTRL + X character within the file.

## Example

If the file PRNT.SUB contains the commands:

```
MOVEF $1 $2  
PIP $3 = $1  
ERA $1
```

and the command format is:

```
SUBMIT PRNT B: *.PRN 7 LST:
```

SUBMIT reads the PRNT.SUB file, substituting B: \*.PRN for all occurrences of \$1, '7' for all occurrences of \$2, and LST: for all occurrences of \$3, resulting in a \$\$\$SUB file containing the commands:

```
MOVEF B: *.PRN 7  
PIP LST :=B: *.PRN  
ERA B: *.PRN
```

which are executed in sequence.

## Using the file STARTUP.SUB

When you startup or reset the system, the operating system automatically enters the command SUBMIT STARTUP. The file STARTUP.SUB may contain the instructions for any initial operations required before using the system. Simply edit the contents of STARTUP.SUB so that it contains the desired commands.

You can use the file START.SUB to automatically startup an application program. For example, the following STARTUP.SUB file initiates the file. SC.COM (Supercalc) located on drive B:

```
USER 2  
B:  
SC
```

# SUBMIT

# SUBMIT

## Error Recovery

1. If it is not necessary to perform initial operations, you may delete the STARTUP.SUB file. This will cause the following 'soft' error each time you reset the system:

Error on Line 001 No 'Sub' File Present

You may ignore this error message.

2. If the number of parameters specified in the command line does not match the number of parameters specified in the .SUB file, SUBMIT terminates with an error message at the console:

INVALID FORMAT:

3. Command processing can be terminated at any time by pressing the BACKSPACE key when the command is read and echoed. In this case, the \$\$\$SUB file is removed, and any commands that follow come from the console.

4. Command processing is also terminated if the operating system detects an error in any of the commands.

5. Programs can terminate the processing of a command file when error conditions occur by simply erasing any existing \$\$\$SUB file.

## Points To Remember

Parameter values inserted in the SUBMIT file cannot be passed as data to a program called by the SUBMIT file unless the XSUB utility has first been called. Refer to the XSUB utility.

The Onyx version of SUBMIT always creates the \$\$\$SUB file on disk A, regardless of the current default disk. File references in the .SUB file should always be preceded by the disk name. (This could be an input parameter. See example.) This eliminates the problem of the operating system not being able to find the correct files.

**Command Formats**

TIME  
TIME <HH> <MM> <SS>

**Tasks Performed**

Sets the system clock to a given time of day or reset the timer to zero when it is being used as an interval timer.

TIME can also be used to display the current time of day or the elapsed time when the clock is in interval timer mode.

**Using the TIME Utility**

The real-time clock can be used in one of two modes: wall-clock mode or interval timer mode. This selection depends on the current tick rate (see the SPEED utility). If the tick rate is 60, then the timer is ticking once every second and is in wall-clock mode. If the tick rate is less than 60, then it is in interval timer mode.

The timer is normally in wall-clock mode and starts up as soon as the operating system is loaded. Until the TIME utility is called to set the clock, the timer can only indicate the hours, minutes and seconds that have elapsed since the operating system was started.

1. To set the current time of day (tick rate = 60),

TYPE:

TIME HH MM SS and press RETURN

- HH is the hours (24 hour clock)
- MM is the minutes
- SS is the seconds

If MM or SS are not entered, they are assumed to be zero.

2. To display the current time,

TYPE:

TIME and press RETURN

If TIME is called without parameters before the wall-clock time has been set, it will display the elapsed time since system start-up.

3. To reset the interval timer to zero in interval timer mode,

TYPE:

TIME 0 and press RETURN

4. To display the current elapsed time in interval timer mode,

TYPE:

TIME and press RETURN

Elapsed time is displayed in the format:

major count / minor count

TIME updates the minor count once every tick, and updates the major count only when the minor count equals the tick rate. For example, if the tick rate is 30, the minor count is updated every 30/60ths of a second (.5 seconds). Whenever the minor count reaches 30, it is set back to zero and one added to the major count.

When elapsed time is displayed, the current tick rate is also displayed.

### Accessing the Timer From a Program

The various components of the timer are stored in the following locations:

OFFD8H	(65496)	major count (3 bytes)
OFFD9H		
OFFDAH		
OFFDBH	(65499)	Number of ticks since last major count update
OFFDCH	(65500)	Current tick rate

The default tick rate is 60 decimal. The actual clock interrupts every 1/60th of a second. Therefore, the major count is incremented by one every 60 ticks, i.e., once a second.

NOTE: The major count is viewed as a single 24 bit wide field — that is, the data is not held 'byte-reversed' as is normally the case with 16 bit numbers.

To convert the major count to wall-clock time:

- Compute (OFFD9H) \* 256 + (OFFDAH); this is the number of seconds in the wall-clock time.
- Divide the number obtained above by the appropriate values of 3600 and 60 as required to convert to hours, minutes and seconds.

### Examples

The following are samples of command formats. Each is entered followed by a RETURN.

1. Set time to 6:56:05 p.m.

TIME 18 56 05

2. Set time to 2:30 p.m.

TIME 14 30 (seconds assumed to be 0)

3. Reset wall-clock time to 0 0 0

TIME 0

The following samples show the display resulting from the command. User input is in **bold**; response follows.

1. Set the clock for use as an interval timer (SPEED must be used to set the appropriate tick rate).

**TIME 0 RETURN**

Real Time Clock reset to 0

2. Display the elapsed time since interval timer set to zero (0).

**TIME RETURN**

Elapsed Time = 0/21 units (Tick is 30/60ths of a second)

There have been 21 minor ticks (at every 1/2 second) in this example since the time was set. There have been 0 major ticks (at one every 15 seconds).

### Error Recovery

TIME displays:

HH MM SS is an illegal time of day

if any of the parameters is out of range.

### Points To Remember

No check is made for midnight and therefore the clock must be reset each day.

### Command Formats

RX  
TX

### Tasks Performed

Transfer files between the Onyx system and another (referred to as the 'other') system running CP/M. TX is the file transmit program, designed to send programs to a system executing RX. RX is the file receive program, designed to receive programs from a system executing TX.

### Using the TX and RX Utilities

Before the TX and RX utilities can be used, they must be installed on the other system. The section Installing TX and RX should be read and understood before proceeding.

TX is designed to work with the program RX running in the destination system. As such, TX is the 'master' program and RX is the 'slave' program. RX receives commands from TX, not the user. For this reason, the user need only worry about console operations for the system in which TX is running.

### Overview

1. Connect the cable between modem port of the Onyx system and null modem port of the other system.
2. Reboot both systems. It is imperative to reboot both systems after connecting the cable since connecting the cable can cause the systems to treat the electrical noise generated as data. This will prevent correct operation of TX.
3. AOA O If using CP/M, set the baud rate on both systems to 9600 baud. Use the SPEED utility for doing this, i.e. SPEED M 9600. This step must **not** be skipped.

To transfer data at some other speed set the baud rates of both systems to this rate (such as 300) and TX and RX will work correctly.

4. Invoke RX on receiving system.
5. Invoke TX on transmitting system.
6. Enter the information that TX asks for — file(s) to transfer and destination disk — and wait for transmission to complete.
7. When transmission is complete (signalled by a BEEP and operator prompt on the transmitting system), press the RETURN key to finish or the 'R' key (repeat) to send more files.

### Invoking the RX Utility

TYPE:

RX and press RETURN

On starting up, RX displays a sign-on message of the form:

FILE RECEIVE UTILITY VERS 3.4 02/09/82

From this point on, unless an error condition occurs, no further operator action is required.

As each data block is received, RX outputs a running count of the data blocks received. At the end of each received file, RX displays the message:

END-OF-FILE RECEIVED

When all files have been received, and on receipt of a command from TX, RX closes down with the message:

LINK TERMINATED

## TX/RX

### Invoking the TX Utility

TYPE:

TX and press RETURN

TX signs on with an identifying message and version number:

```
FILE TRANSFER UTILITY VERS 3.5 04/09/82
HIT CR WHEN RECEIVE MACHINE READY OR Q
TO ABORT
```

PRESS:

RETURN

TX attempts (by sending out the letter 'S', which is displayed, and waiting for a reply) to link up with the RX program.

If the link can be established, TX displays the message:

```
LINK TO SLAVE MACHINE ESTABLISHED
```

But if ten attempts fail to establish the link, TX displays:

```
UNABLE TO ESTABLISH/MAINTAIN DATA LINK
```

If the link is established, TX asks for the name(s) of the files to be transferred:

TYPE:

filename and press RETURN

- filename is a possibly ambiguous filename (using '?' only) optionally preceded by a disk name.

TX next prompts:

```
DESTINATION DISK (CR for current):
```

PRESS:

RETURN

for the current default disk.

or

TYPE:

disk

- disk is the name of the disk on the other system where the files are to be stored.

## TX/RX

Finally TX asks:

```
DESTINATION FILE NAME:
```

TYPE:

filename and press RETURN

- filename is a possibly ambiguous filename. The ambiguity must match **exactly** the ambiguity of the source filename.

TX then displays the unambiguous names of all files to be transferred.

As a file is being transferred, TX displays its file name. Both TX and RX display a record count. This serves to indicate that data is being transmitted and received. It is normal to see a difference of one record between the two counts.

When the file(s) have been transmitted, TX displays the message:

```
FUNCTION COMPLETE
TYPE R TO REPEAT, CR TO EXIT
```

If more file(s) are to be transferred,

PRESS:

R

and TX requests file names as before.

PRESS:

RETURN

to cause TX to command RX to shut down. TX will also shut down.

### Error Recovery

If the data link cannot be established or maintained, it is necessary to reset the receiving system in order to exit from RX. Possible error messages from TX are shown in Table 3-7.

Messages	Messages
<p>1. FILE ALREADY EXISTS ON RECEIVING MACHINE: TX does not destroy an existing file. The existing file may be renamed and tried again if desired. As an expediency, send the file again but with a temporary destination file name.</p>	<p>9. CAN'T COMPLETE "OPEN FILE" COMMAND: Couldn't open file on receiving system for a reason other than 'file already exists'. Make sure that the directory isn't filled up.</p>
<p>2. ERROR IN ATTEMPTING TO OPEN SOURCE FILE: The source file probably does not exist.</p>	<p>10. BUF NOT RECEIVED CORRECTLY — RETRYING: The actual data's checksum was wrong and will be sent again. This message is only a warning and is not fatal.</p>
<p>3. UNABLE TO ESTABLISH/MAINTAIN DATA LINK: If this happens during initial SYNC it is likely the systems are not wired together correctly or are at different baud rates. If it happens at other times a bit was probably dropped.</p>	<p>11. CAN'T SEND FILE RECORD CONTENTS CORRECTLY: After repeated attempts, TX cannot transmit the file name or the file data correctly. Indicates a dropped bit, hardware problems, or causes discussed in #3 and #4.</p>
<p>4. HARD DATA TRANSMISSION ERROR: As each data block is sent, a checksum is calculated and transmitted. If RX detects a checksum failure on recomputing the checksum, then it requests that TX retransmit the data block. This retransmission is repeated a predetermined number of times. If the clock cannot be transmitted successfully, then TX displays this message. Data was lost and maximum number of retries were done. Possible causes of this failure are a loose plug or bad EIA cable. See also #3 for causes.</p>	<p>12. CAN'T COMPLETE "WRITE RECORD" COMMAND: Retries failed on command request.</p>
<p>5. RECEIVER HAS UNRECOVERABLE DISK ERROR: Possible causes are insufficient disk space or directory space on receiving system.</p>	<p>13. CAN'T COMPLETE "SENDING NEXT RECORD" COMMAND: Retries failed on command request.</p>
<p>6. RECEIVER CANNOT CLOSE FILE: The disk on which RX attempts to place the incoming data file is Write Protected, or there is insufficient space on the disk.</p>	<p>14. CAN'T COMPLETE "CLOSE FILE" COMMAND: Retries failed on command request.</p>
<p>7. CAN'T COMPLETE "ESTABLISH LINK" COMMAND: See #3 for causes.</p>	<p>15. MISSING ': FOLLOWING DESTINATION DISK LETTER: To ask that the file(s) be copied to a different disk on the receiving system (than on the transmitting system), type the letter of the disk followed by a ':.</p>
<p>8. CAN'T SEND FILENAME CORRECTLY: The checksum on the transferred filename was wrong and retries failed. Indicates a dropped bit, an error in following this procedure, or hardware problems.</p>	<p>16. UNRECOVERABLE ERROR: The previous error reported is fatal and TX is terminating.</p>
	<p>17. ATTEMPT TO ESTABLISH LINK FAILED — RETRYING: An attempt by the transmitting system to initially synchronize failed and will be attempted again. Usually if this message appears then either it will repeat until the retry count is used up (resulting in TX terminating) or a fatal error will occur later, usually in sending the filename. The solution is to reboot both systems and set their baud rates.</p>

## **TX/RX**

If the operation procedure specified here is followed with a direct wire connection then the TX/RX file transfer utility runs very reliably and trouble-free. Most problems experienced with TX are actually either:

1. The cable is not connected to the right port.
2. There isn't a null modem between the systems (i.e., pins 2 and 3 are not swapped.)
3. The transmitting and receiving systems are set to different baud rates.
4. A hardware problem.
5. The baud rate is too high. Try a slower rate.

### **Points To Remember**

1. If both systems are not rebooted after the cables are installed then random characters, which TX and RX are not designed to cope with, may be received, resulting in unpredictable results.
2. If the modem SPEED is not set on the Onyx system then the modem port may be in an undefined state despite **SPEED M ?** saying that the speed is 9600 baud and all is well.
3. If TX retries the initial SYNC (link command S) many times and fails it is probable that the baud rate of the receiving system is lower than the baud rate of the transmitting system or that both systems were not rebooted and SPEED set after connecting the cables.
4. If TX stops in the middle of the initial SYNC then it is probable that either the baud rate of the receiving system is too high or that the cable is not connected correctly or that the receiving system is not listening to the transmitting system.

## **TX/RX**

### **Installing TX and RX**

#### **User Knowledge Assumed**

The user should be familiar with CP/M itself and also with the standard CP/M utilities DDT, PIP, and ASM. These utilities are used to set the environment for TX/RX to operate.

#### **Assumptions about the Other Computer System**

It is assumed that the other computer system has a serial port other than that used for its console and that the parameters of this serial port (the port number for status and control, and the status bit significance) are known by the user. Refer to the appropriate Onyx system hardware manual for its serial port parameters.

#### **Configuring TX/RX for Onyx Systems**

TX and RX have been configured for the Onyx system for which it was built. No changes are necessary. These programs are TX.COM and RX.COM.TX.ASM and RX.ASM have been included so that they may be configured for the other system.

#### **Establishing the Hardware Connection**

Although the most likely operation to be performed is that of TXing programs from the other system to the Onyx system, it is easier to establish the communications link between the two systems by first getting PIP to send a copy of TX and RX source code to the other system.

However, before this can be attempted, the Onyx system must establish electrical communication with the other system. There are two hardware considerations: the speed (baud rate) of the communication and the correct physical connection (cable) between the two systems.

In order for the other system to receive data correctly, the baud rates of the two serial ports (on the Onyx and on the other system) must be identical. The SPEED utility is used to set the baud rate on the Onyx System.

## TX/RX

The physical wiring of the EIA cable used to connect the two systems must be such that the Transmitted Data of one system is the Received Data of the other. In other words, pin 2 of the Onyx System port (either serial printer or modem) must be connected to pin 3 of the corresponding port on the other system, and pin 2 of the other system must be connected to pin 3 of Onyx system port (either serial printer or modem). Status lines such as DTR, DSR, RTS, ETS may also have to be considered.

### Establishing the Software Connection

#### 1. Modifying PIP to Receive Data on the Other System

The best method for ensuring that the two systems can communicate with each other is to use PIP and communicate via the **serial printer port**. A special version of PIP (PIPRX) is built for the other system to receive files. PIP (unmodified) is used on the Onyx system to transmit the files. In this way RX.ASM and TX.ASM can be transferred to the other system.

This presupposes that the user has the appropriate hexadecimal values for:

- The Serial Data Port
- The Serial Status Port
- The mask for the Data Ready status, and knowledge of whether a bit is set high or low to indicate an incoming data character.

The procedure for building PIPRX for the other system is illustrated by the following console dialogue. Comments are shown in '( )'.

## TX/RX

### DDT PIP.COM RETURN

DDT 2.0

NEXT PC

01E00 0100

-A103

0103 CALL 110 RETURN

0105 RETURN

-A110 RETURN

0110 IN xx RETURN

(xx = hex status port number)

0112 ANI yy RETURN

(yy = hex mask for Data Ready)

0114 JZ 110 RETURN

(JNZ if Data Ready bit = 0  
when Data there)

0117 IN zz RETURN

(zz = hex data port number)

0119 ANI 7F RETURN

(strip parity)

011B STA 109 RETURN

(for PIP)

011E RET RETURN

(return to PIP)

011F RETURN

-CTRL + C

(return to CP/M)

A> SAVE 29 PIPRX.COM (Save new version of PIP)

#### 2. Sending RX.ASM and TX.ASM to the Other System

This new version of PIP accepts data coming in on the specified serial port when the device name filename 'INP:' is used.

On the other system,

TYPE:

PIPRX RX.ASM=INP:(E) and press RETURN

to read data from the serial port into the file RX.ASM. The (E) option indicates that each character received is to be displayed on the console. This allows PIPRX to be used as a line monitor to show the incoming data.

On the Onyx system,

TYPE:

PIP LST:=RX.ASM,EOF: and press RETURN

NOTE: The only method of terminating PIPRX (other than when it decides to terminate operation) is by resetting the other system. It normally takes several attempts to get the two systems to 'talk' to each other. Persevere!

## TX/RX

When the complete source code for RX.ASM has been received by PIPRX, it should close the file RX.ASM and then terminate. If this does not happen, the condition can be corrected by connecting a terminal to the other system's serial port. When the terminal is set to run at the baud rate of the port,

PRESS:

CTRL + Z

(hold down CTRL while pressing Z). When PIPRX sees this, it closes the file and terminates.

Repeat this operation for TX.ASM.

### 3. Configuring TX/RX for the Other System

Generally, the only EQUate variables that need changing are STAT, DATA, TBE, and DRDY. STAT is the modem port's STATUS register I/O port number. DATA is likewise for the modem port's DATA register I/O port number. TBE is a mask of the bit of the STATUS register that becomes true when the byte being sent to the port has been transmitted and the port is now ready to transmit another byte, and DRDY is a mask of the bit that becomes true when data has been received.

The C8000 and C5000 variables control code for the Onyx systems. It may have to be modified in order to run on the other system. It may also be necessary to change the code in the SEND and RECEIVE routines. In particular the code that tests the status of the I/O port for this hardware may need modification.

The sign-on message for this installation and for the baud rate if other than 9600 needs to be changed. If it is to be on a CP/M lookalike a few other routines may need to change if it doesn't emulate CP/M exactly.

It is recommended that the style that Onyx has used for conditional code, complete with true-false conditional variables for each parameter, be adhered to.

## TX/RX

To use the same code on a variety of systems it is recommended that an equate variable exist for each one that will be true if code is to be assembled for that system. Onyx has done this with conditional assembly code (variables C8000, C5000, etc.) to handle their individual needs. This will also enable easy upgrade to later releases of TX/RX.

Next, the edited versions of RX and TX must be assembled. There should be no assembly errors. The LOAD command is used to generate the .COM files.

### PIPRX on the Onyx System

If it is desired to use the PIP and PIPRX combination to transfer files from the other system to the Onyx system, then PIP must be modified to create a PIPRX for the Onyx. The version of PIPRX that is included on the SDT differs from the PIPRX for the other system in the following ways:

- PIPRX on the Onyx system uses the **modem port** rather than the serial printer port to send and receive data.
- The instructions patched into PIPRX are completely different from those shown in the example above. Rather than reading an I/O port directly, they make a direct call into the BIOS (Basic I/O System) to read the modem port. A CALL must be effected using the PCHL instruction.

Specifically, the instructions that have been inserted at location 110 are:

```
LXI    H,011C    ; Fake return link on stack
PUSH   H
LXI    H,0015    ; Build BIOS address
LDA    2         ; Base page of BIOS vector
MOV    H,A
PCHL                      ; 'CALL' Modem Input
STA    109       ; Store returned character
RET                                ; Return to PIP
```

**TYPE Display ASCII Text File on Console****Command Format**

TYPE <filename>

**Tasks Performed**

TYPE displays the contents of an ASCII source file on the console.

**Using the TYPE Command**

TYPE:

TYPE filename and press RETURN

- filename is an unambiguous file reference

The TYPE command expands tabs (CTRL + I characters), assuming tab positions are set at every eighth column. The filename may be preceded by an optional disk name.

**Examples**

The following are samples of command formats. Each is entered followed by a RETURN.

1. Display a file located on the current default disk.

TYPE X.Y

2. Display a file located on a disk other than the current default.

TYPE B:X.PRN

The file X.PRN from disk B is displayed.

**Error Recovery**

If TYPE cannot find the specified file it displays the filename followed by a '?'.  
(

## Command Format

UNERA <filename>

## Tasks Performed

UNERA recovers a file which has been inadvertently ERAsed.

## Using the UNERA Utility

To recover a file which has been erased,

TYPE:

UNERA filename and press RETURN

- filename is a possibly ambiguous file name.

## Error Recovery

UNERA displays:

Erased File filename not found

if the file does not exist.

## Points To Remember

**WARNING:** If a warm boot or any file operation is performed between the ERA command and the UNERA command, it is possible that the directory will be corrupted with disk blocks being assigned to more than one file. It is possible that some or all of the files on the disk may be lost. To prevent this after UNERAing the file, it should be PIPed to another disk, reERAsed, and the system restarted (warm or cold boot) to maintain file system integrity.

There should be only one user on the system when running UNERA under MP/M.

## **USER** Set Current User Directory Number

## **USER**

### **Command Format**

USER <n>

### **Tasks Performed**

Changes user directory numbers.

### **Using the USER Command**

TYPE:

USER n and press RETURN

- n is an integer ranging from 0 to 15.

### **Example**

The following sample command format requests a change from the current user directory to directory 4.

USER 4

### **Error Recovery**

USER responds with a question mark to an invalid user number.

### **Points To Remember**

The user directory number is valid for all disks on the system.

When the system is powered on or reset (cold boot), the user directory number is set to 0.

The current user directory number is displayed after each warm boot unless disabled by the OPTIONS utility.

To determine the current user number, use the STAT Utility

STATUSR:

or

PRESS:

CTRL + C

(hold down CTRL while pressing C).

## **VIS Remove SYS and R/O Status From File(s)**

**VIS**

### **Command Format**

VIS <filename> {user}

### **Tasks Performed**

VIS sets a file's status to DIR and R/W to allow it to be copied with PIP (DIR status) or to be ERAsed (R/W status).

### **Using the VIS Utility**

1. To set the status of a file in the current directory to DIR and R/W,

TYPE:

VIS filename and press RETURN

- filename is a possibly ambiguous filename.

2. To set the status of a file in any directory,

TYPE:

VIS filename user and press RETURN

- user is the user directory number which is to be searched for the file(s).

If the file(s) are found, DIR and R/W status will be set.

### **Example**

The following are sample command formats. Each is entered followed by a RETURN.

1. Set DIR and R/W status for file DATA.COM in user directory 5.

VIS DATA.COM 5

2. Set DIR and R/W status for all .COM files on disk C in the current directory.

VIS C:\*.COM

### **Error Recovery**

If the file cannot be found, VIS displays:

File filename not found on user n.

### **Points To Remember**

There should be only one user on the system when running VIS under MP/M.

## **VTSETUP** Configure CP/M to Switch Sundance or Onyx Terminal to 80 Column ANSI Mode on Warm Boot

## **VTSETUP**

### **Command Format**

VTSETUP

### **Tasks Performed**

VTSETUP sets a flag in the operating system so that the Onyx terminal or the Sundance is put in ANSI mode whenever a warm boot occurs. In addition, the 132 column versions are put in 80 column mode.

### **Using the VTSETUP Utility**

TYPE:

VTSETUP and press RETURN

### **Error Recovery**

None needed.

### **Points To Remember**

VTSETUP needs to be run each time the system is reset (cold boot or power on). The easiest way to accomplish this is to include VTSETUP in the STARTUP.SUB file.

**CAUTION:** Do not use VTSETUP with MP/M unless all terminals are Sundances or Onyx terminals.

## **VT52 Switch Sundance or Onyx Terminal from ANSI Mode to VT52 Mode**

## **VT52**

### **Command Format**

VT52

### **Tasks Performed**

VT52 switches the Onyx terminal or the Sundance from ANSI mode to VT52 mode.

### **Using the VT52 Utility**

TYPE:

VT52 and press RETURN

VT52 should be run prior to running any application which uses VT52 mode commands such as those controlling the cursor. See the Terminal Reference Manual for a complete list of VT52 commands.

### **Error Recovery**

None needed.

### **Points To Remember**

If VTSETUP has been run, the Onyx terminal or Sundance returns to ANSI mode automatically if a warm boot occurs after the application terminates.

**Command Format**

XSUB

**Tasks Performed**

Extends the capability of the SUBMIT utility to allow input parameters to be passed to programs which read buffered console input.

**Using the XSUB Utility**

XSUB is included in the SUBMIT file as the first line. With XSUB as the first line, the SUBMIT file may now contain lines of parameterized (\$1, \$2...etc.) data which can be passed directly to a utility invoked by the SUBMIT file.

Refer to the SUBMIT Utility.

**Example**

If the SUBMIT file DDTEST.SUB has the following commands:

```
XSUB
DDT
I $1.HEX
R
G
SAVE $2 $3.COM
```

and the command line is:

```
SUBMIT DDTEST TEST 5 NEWTEST
```

The actual commands implemented are:

```
DDT
I TEST.HEX
R
G
SAVE 5 NEWTEST.COM
```

**Error Recovery**

XSUB has no error messages. Refer to the SUBMIT utility for error messages which may occur during the processing of the submit file.

**Points To Remember**

XSUB remains in memory until processing of the SUBMIT file is complete. After each warm boot which occurs during the processing of the submit file XSUB displays:

(xsub active)

## 3.4 HARDWARE ERROR MESSAGES

### Disk Error Messages

Whenever the disk driver detects an error condition, it displays the message

**\*\*\*DISK ERROR disk, code, track, sector**

The various codes are shown in Table 3-8. To recover, try to load the operating system into memory.

If the error is the one marked by a '+' and it persists, use the SPARE utility to spare the sector causing the error. Track and sector numbers are used by the SPARE utility when sparing a bad sector. Refer to the SPARE utility on their use.

If any other errors persist, reload the utilities from a backup tape or the operating system and utilities from the Software Distribution Tape.

If either process is unsuccessful at eliminating the errors, run the Disk Exerciser described in Other SDT Functions (Section 3.5); then have the system serviced. Record all error messages to aid the service technician in isolating the problem.

### Tape Error Messages

Tape error messages may be displayed when one of the tape cartridge utilities — DBACK, DLOAD, DVFY, ARCHIVE — is running. The error message has the format:

Tape Error nn - descriptive message

Try, if possible, to use another tape and make sure the tape in use has been properly initialized. Refer to the appropriate utility in Section 3.3. See also the Hardware Operations Manual or User Guide for details on tape initialization.

Tape error numbers greater than 27 display the message

Unassigned Error Code

and usually result from trying to write on a tape that has been protected (made 'SAFE' by turning the pointer towards 'SAFE'). Check the tape and restart the utility.

Table 3-8. Disk Error Messages

Error Code	Message	Error Code	Message
64	No acknowledge from disk	81	Unallocated
65	Incorrect busy acknowledge	82	Fault status after read
66	Timeout error	*83	Sector data CRC error
67	Fault status in drive	*84	Bad sector header information
+68	CRC Error	85	Write on protected drive
6A	Verify error	86	Sector number out of range
78	Error in preparing for read/write	87	Spare directory full
79	Timeout error	88	Write error on spare directory
7A	Fault status during seek	89	Attempt to spare (yank) a sector already spared
7B	Seek error		
*7C	Sector header CRC error		
7D	Home to track 0 timeout error		
7E	Fault after home to track 0		
7F	Drive not ready (on-line)		
80	Fault status after write		

A '\*' marks the errors which may be overcome by reloading the disk from a backup tape.

A '+' marks the errors which may be overcome by sparing the sector.

**Table 3-9. Tape Error Messages**

<b>Error Code</b>	<b>Explanation</b>	<b>Message</b>
1	LOPENT was not called	Tape not open.
2	End of tape cartridge reached	Unexpected end of tape cartridge reached.
3	Blocksize error	Illegal blocksize.
4	Error rewinding tape	Error while rewinding tape.
5	Tape already open	Attempt to re-open tape.
6	Illegal channel	Illegal channel specified.
7	Could mean illegal channel requested or drive ready	Tape drive not on line.
8	Drive not read - Tape probably not inserted.	Tape drive not ready.
9	Can't write on tape - Open as read only	Tape cartridge write-protected ('SAFE').
10	End of tape reached	Requested block beyond end of tape.
11 - 17	Tape write error	Fatal error while writing tape.
18	Rewind error while writing Tape	Rewind error while writing tape.
19	Non-retryable error in readata	Fatal error while reading tape.
20	Retry count exceeded in readata	Retry count exceeded while reading tape.
21	Seek error in readata	Seek error while reading tape.
22 - 24	Several errors while backing up tape	Unable to back up while reading tape.
26	Two attempts to error move	Tape motion error.
27	Probably an illegal block number.	Unable to position tape to requested block.

## 3.5 OTHER SDT FUNCTIONS

---

In addition to the operating system and its utilities, the Software Distribution Tape (SDT) contains a number of other programs which are used to perform system maintenance and diagnostic functions. These include:

- Software Load—recreate the system disk
- Change the interlace factor of the current disk (normally never done)
- Display the software serial number

### Using the SDT

Insert the SDT tape in the tape drive and press the reset button located in the back of the system. Refer to the appropriate Hardware Operations Manual or User's Guide for the location.

When the '\*' appears,

TYPE:

T

The following menu displays:

\*\*\*\* ONYX SOFTWARE DISTRIBUTION TAPE \*\*\*\*

1. Set disk (current = 0).
2. RECREATE system disk.
3. Display delivered software directory.
4. Recreate Alternative system.
5. Set sector interlace.
6. Return to prom monitor.
7. Recreate reserved area.
8. FORMAT the disk.

Request:

The options perform as follows:

1. To set disk (current = 0):

TYPE:

1 and press RETURN

Changing the disk code makes it possible to manipulate an external disk as if it were the disk in the system enclosure. That is, after using this option, you may RECREATE an external disk and reset the interlace factor.

The system prompts:

Drive: (0-7):

TYPE:

n and press RETURN

- n if for the internal disk; n = 1, 2 ... is an external disk.

2. To RECREATE system disk:

TYPE:

2 and press RETURN

This option loads the operating system and utilities onto the disk selected with option 1. If you select an external disk with option 1, the software loads onto the first logical drive of that disk.

The following message appears on the screen:

—WARNING—RECREATING drive 0 will destroy the disk contents!

A 'y' recreates disk A and then returns to the menu. A 'n' stops this operation and returns to the menu.

3. To display the software directory:

TYPE:

3 and press RETURN

The software directory shows the revision number of the operating system on the tape, the system it applies to, and when it was made.

4. To recreate alternate system:

TYPE:

**4** and press **RETURN**

This option allows you to recreate the system disk with the alternate copy of the software. (Two copies of the software have been provided on the SDT). Use this option if there is a problem using the copy of the software loaded with option 2.

This option loads the operating system and utilities onto the disk selected with option 1. If option 1 selects an external disk, the software loads onto the first logical drive of that disk.

5. To set sector interlace (current = 5:1)

TYPE:

**5** and press **RETURN**

The interlace factor has been optimally set for each disk type at the factory and is normally never changed. For the Sundance, C5001/SU and C8001/SU the interlace is set at 5:1. C5001/MU and C8001/MU systems have an interlace factor of 7:1. An external disk may have a different interlace factor than the disk in the system enclosure. The message reads:

Interlace (1-19):

**WARNING:** Do not change the interlace factor if there are files on the disk which need to be saved. Back up all files (on a file by file basis) before changing the interlace factor. Once changed, erase the disk to clear all directory entries before restoring the files.

TYPE:

**n** and press **RETURN**

- n is a number between 1 and 19.

6. To return to prom monitor:

TYPE:

**6** and press **RETURN**

Option 6 allows direct return to the prompt (\*) without pressing the reset button.

7. To recreate the reserved area only

TYPE:

**7** and press **RETURN**

This option allows you to re-create the reserved area without destroying the first logical drive. The utilities are not loaded. If you are using a SDT with MP/M, the operating system is not loaded, only the MP/M loader is loaded. This option can be used if the drive capacity given by the operating system is completely disparate with the actual size.

8. To FORMAT the disk

TYPE:

**8** and press **RETURN**

Instructions for using this option are contained in the "DISK INT Addendum" to your operating system Software Release Notice.

**WARNING:** FORMATTING will destroy the disk contents! The spare table will also be destroyed and rebuilt. The Formatter program should be run for several hours to obtain a reasonable confidence level in the integrity of the disk.

## 3.6 STANDARD 7-BIT ASCII CHARACTER SET

Table 3-10. Standard 7-Bit ASCII Character Set

Dec	Hex	Oct	Binary	Key	Definition
0	00	000	0000 0000	CTRL/@	NUL Filler
1	01	001	0000 0001	CTRL/A	SOH Start of Heading
2	02	002	0000 0010	CTRL/B	STX Start of Text
3	03	003	0000 0011	CTRL/C	ETX End of Text
4	04	004	0000 0100	CTRL/D	EOT End of Transmission
5	05	005	0000 0101	CTRL/E	ENQ Enquiry
6	06	006	0000 0110	CTRL/F	ACK Acknowledge
7	07	007	0000 0111	CTRL/G	BEL Bell
8	08	010	0000 1000	CTRL/H	BS Backspace
9	09	011	0000 1001	CTRL/I	HT Horizontal Tabulation
10	0A	012	0000 1010	CTRL/J	LF Line Feed
11	0B	013	0000 1011	CTRL/K	VT Vertical Tabulation
12	0C	014	0000 1100	CTRL/L	FF Form Feed
13	0D	015	0000 1101	CTRL/M	CR Carriage Return
14	0E	016	0000 1110	CTRL/N	SO Shift Out
15	0F	017	0000 1111	CTRL/O	SI Shift In
16	10	020	0001 0000	CTRL/P	DLE Data Link Escape Cursor Address
17	11	021	0001 0001	CTRL/Q	DC1 Device Control 1
18	12	022	0001 0010	CTRL/R	DC2 Device Control 2
19	13	023	0001 0011	CTRL/S	DC3 Device Control 3
20	14	024	0001 0100	CTRL/T	DC4 Device Control 4
21	15	025	0001 0101	CTRL/U	NAK Negative Acknowledge
22	16	026	0001 0110	CTRL/V	SYN Synchronous Idle
23	17	027	0001 0111	CTRL/W	ETB End of Transmission Block
24	18	030	0001 1000	CTRL/X	CAN Cancel
25	19	031	0001 1001	CTRL/Y	EM End of Medium
26	1A	032	0001 1010	CTRL/Z	SUB Substitute
27	1B	033	0001 1011	CTRL/〔	ESC Escape
28	1C	034	0001 1100	CTRL/\\	FS File Separator
29	1D	035	0001 1101	CTRL/〕	GS Group Separator
30	1E	036	0001 1110	CTRL/^	RS Record Separator
31	1F	037	0001 1111	CTRL/.	US Unit Separator

Dec	Hex	Oct	Binary	Key	Definition
32	20	040	0010 0000	Space Bar	SP Space
33	21	041	0010 0001	!	Exclamation Mark
34	22	042	0010 0010	"	Double Quote
35	23	043	0010 0011	#	Number Symbol
36	24	044	0010 0100	\$	Dollar Symbol
37	25	045	0010 0101	%	Percent Symbol
38	26	046	0010 0110	&	And - Ampersand
39	27	047	0010 0111	'	Single Quote
40	28	050	0010 1000	(	Left Parenthesis
41	29	051	0010 1001	)	Right Parenthesis
42	2A	052	0010 1010	*	Asterisk
43	2B	053	0010 1011	+	Plus Symbol
44	2C	054	0010 1100	,	Comma
45	2D	055	0010 1101	-	Minus/Hyphen Symbol
46	2E	056	0010 1110	.	Period/Decimal Point
47	2F	057	0010 1111	/	Divide Symbol - Slash/Slant
48	30	060	0011 0000	0	Zero
49	31	061	0011 0001	1	One
50	32	062	0011 0010	2	Two
51	33	063	0011 0011	3	Three
52	34	064	0011 0100	4	Four
53	35	065	0011 0101	5	Five
54	36	066	0011 0110	6	Six
55	37	067	0011 0111	7	Seven
56	38	070	0011 1000	8	Eight
57	39	071	0011 1001	9	Nine
58	3A	072	0011 1010	:	Colon
59	3B	073	0011 1011	;	Semicolon
60	3C	074	0011 1100	<	Less Than Symbol - Left Caret
61	3D	075	0011 1101	=	Equal Symbol
62	3E	076	0011 1110	>	Greater Than/Right Caret
63	3F	077	0011 1111	?	Question Mark
64	40	100	0100 0000	@	At Sign - Ampersand
65	41	101	0100 0001	A	
66	42	102	0100 0010	B	

Dec	Hex	Oct	Binary	Key	Definition
67	43	103	0100 0011	C	
68	44	104	0100 0100	D	
69	45	105	0100 0101	E	
70	46	106	0100 0110	F	
71	47	107	0100 0111	G	
72	48	110	0100 1000	H	
73	49	111	0100 1001	I	
74	4A	112	0100 1010	J	
75	4B	113	0100 1011	K	
76	4C	114	0100 1100	L	
77	4D	115	0100 1101	M	
78	4E	116	0100 1110	N	
79	4F	117	0100 1111	O	
80	50	120	0101 0000	P	
81	51	121	0101 0001	Q	
82	52	122	0101 0010	R	
83	53	123	0101 0011	S	
84	54	124	0101 0100	T	
85	55	125	0101 0101	U	
86	56	126	0101 0110	V	
87	57	127	0101 0111	W	
88	58	130	0101 1000	X	
89	59	131	0101 1001	Y	
90	5A	132	0101 1010	Z	
91	5B	133	0101 1011	(	Left Bracket
92	5C	134	0101 1100	\	Back Slash/Slant
93	5D	135	0101 1101	)	Right Bracket
94	5E	136	0101 1110	^	ASCII Up Arrow - Circumflex
95	5F	137	0101 1111	_	Underscore
96	60	140	0110 0000	`	Back Quote
97	61	141	0110 0001	a	
98	62	142	0110 0010	b	
99	63	143	0110 0011	c	
100	64	144	0110 0100	d	
101	65	145	0110 0101	e	

Dec	Hex	Oct	Binary	Key	Definition
102	66	146	0110 0110	f	
103	67	147	0110 0111	g	
104	68	150	0110 1000	h	
105	69	151	0110 1001	i	
106	6A	152	0110 1010	j	
107	6B	153	0110 1011	k	
108	6C	154	0110 1100	l	
109	6D	155	0110 1101	m	
110	6E	156	0110 1110	n	
111	6F	157	0110 1111	o	
112	70	160	0111 0000	p	
113	71	161	0111 0001	q	
114	72	162	0111 0010	r	
115	73	163	0111 0011	s	
116	74	164	0111 0100	t	
117	75	165	0111 0101	u	
118	76	166	0111 0110	v	
119	77	167	0111 0111	w	
120	78	170	0111 1000	x	
121	79	171	0111 1001	y	
122	7A	172	0111 1010	z	
123	7B	173	0111 1011	{	Left Brace
124	7C	174	0111 1100		Vertical Bar
125	7D	175	0111 1101	}	Right Brace
126	7E	176	0111 1110	~	Tilde
127	7F	177	0111 1111	DEL	RUBOUT

## 3.7 SUBJECT INDEX

## SUBJECT INDEX

<b>Command Operations</b>	<b>Page</b>
DO .....	3-30
OPTIONS .....	3-51
SUBMIT .....	3-73
XSUB .....	3-89

### **Disk Directory Operations**

DIR .....	3-27
DSTAT .....	3-31
FINDALL .....	3-43
INVIS .....	3-45
MOVEF .....	3-49
PUB/PRIV .....	3-61
REN .....	3-63
STAT .....	3-69
USER .....	3-85
VIS .....	3-86

### **Disk Maintenance**

ARCHIVE .....	3-9
DBACK .....	3-22
DDUMP .....	3-25
DLOAD .....	3-28
DSTAT .....	3-31
LOADREL .....	3-47
SPARE .....	3-65
STAT .....	3-69

<b>File Copy</b>	<b>Page</b>
ARCHIVE .....	3-9
FBACK .....	3-11
FLOAD .....	3-11
LOADREL .....	3-47
PIP .....	3-54
PIPRX .....	3-81
TX/RX .....	3-77

### **File Display**

DDUMP .....	3-25
FCOMP .....	3-39
FDUMP .....	3-40
FILESET .....	3-41
PIP .....	3-54
TYPE .....	3-83

### **File Erase/Recover**

ERA .....	3-37
ERASEALL .....	3-38
QERA .....	3-62
UNERA .....	3-84

### **File Modify**

DDT .....	3-24
ED .....	3-35
PIP .....	3-54

## SUBJECT INDEX

<b>File Operations</b>	<b>Page</b>
ARCHIVE .....	3-9
FBACK .....	3-11
FLOAD .....	3-11
DDT .....	3-24
DUMP .....	3-32
ED .....	3-35
ERA .....	3-37
ERASEALL .....	3-38
FILESET .....	3-41
FINDALL .....	3-43
INVIS .....	3-45
LOADREL .....	3-47
MOVEF .....	3-49
PIP .....	3-54
PIPRX .....	3-81
PUB/PRIV .....	3-61
QERA .....	3-62
REN .....	3-63
SAVE .....	3-64
STAT .....	3-69
TX/RX .....	3-77
UNERA .....	3-84
VIS .....	3-86

<b>File Status</b>	<b>Page</b>
INVIS .....	3-45
MOVEF .....	3-49
PUB/PRIV .....	3-61
REN .....	3-63
STAT .....	3-69
VIS .....	3-86

<b>Peripheral Control</b>	<b>Page</b>
ASSIGN .....	3-17
FREEZE .....	3-44
PROTOCOL .....	3-59
SPEED .....	3-67

## SUBJECT INDEX

<b>Program Operations</b>	<b>Page</b>
ASM .....	3-16
CCREADY .....	3-19
DDT .....	3-24
ED .....	3-35
LOAD .....	3-46
SAVE .....	3-64

<b>System Operations</b>	<b>Page</b>
ASSIGN .....	3-17
DATE .....	3-21
DSTAT .....	3-31
FREEZE .....	3-44
OPTIONS .....	3-51
PATCH .....	3-53
PUB/PRIV .....	3-61
SAVE .....	3-64
SPEED .....	3-67
STAT .....	3-69
TIME .....	3-75
VTSETUP .....	3-87

<b>Tape Operations</b>	<b>Page</b>
ARCHIVE .....	3-9
FLOAD .....	3-11
FBACK .....	3-11
TVFY .....	3-11
TFORM .....	3-11
END .....	3-11
DBACK .....	3-22
DLOAD .....	3-28
DVFY .....	3-33
LOADREL .....	3-47

<b>Terminal Control</b>	<b>Page</b>
ANSI .....	3-8
COL80/COL132 .....	3-20
VTSETUP .....	3-87
VT52 .....	3-88

