

Getting Started with **RTE-6/VM**

HP 1000 Computer Systems



HP Computer Museum

www.hpmuseum.net

For research and education purposes only.

Getting Started with RTE-6/VM

HP 1000 Computer Systems



HEWLETT
PACKARD

HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014

MANUAL PART NO. 92084-90002
Printed in U.S.A. December 1981

Printing History

The Printing History below identifies the Edition of this Manual and any Updates that are included.

First Edition Dec 1981

NOTICE

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

HEWLETT-PACKARD 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. Hewlett-Packard 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.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

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 Hewlett-Packard Company.

Copyright © 1981 by HEWLETT-PACKARD COMPANY

Preface

This manual is a tutorial guide for the first-time user of the HP 1000 System. It is an introduction to the RTE-6/VM Operating System with Session Monitor. This manual shows you how to:

- a. Communicate with the system.
- b. Create, modify, and manipulate files.
- c. Develop and run programs.
- d. Get help in case of difficulties.

Try each example as you read the sections. Show here is only a small subset of the commands and capabilities available with RTE-6/VM. As you become more familiar with the system, refer to the RTE-6/VM manuals for further information. The index manual (RTE-6/VM Index to Operating System Manuals, 92084-90001) provides a documentation map and description of each manual.

CONTENTS

Chapter 1	Page
TALKING TO THE SYSTEM	
The Terminal	1-1
Log-On Identification	1-2
Password	1-2
Capability Level	1-2
Log-On Procedure	1-2
Error Recovery	1-4
Interacting With the System	1-4
Break Mode	1-5
The File Manager Program	1-6
Log-Off Procedure	1-6

Chapter 2	Page
USING DISC FILES	
Files	2-1
File Names and File Descriptors	2-1
Where Files are Stored	2-2
Disc Cartridges	2-2
Private Group and System Cartridges	2-3
Accessing Cartridges	2-3
Logical Unit Number	2-3
Cartridge Tape Drive	2-3
Cartridge Reference Number	2-3
Cartridge List	2-4
File Directory	2-4
Creating Files	2-5
Changing a File Name	2-6
Deleting Files	2-6

Chapter 3	Page
USING THE EDITOR	
Modifying Files	3-1
Start Screen Mode	3-2
Screen Mode Commands	3-2
Line Edits	3-3
Character Edits	3-6
Saving the Modified File	3-7

Chapter 4	Page
PROGRAM DEVELOPMENT	
Writing the Source Program	4-2
Compile and Load a Program	4-3
Implied Run	4-6
Saving a Program	4-6

Chapter 5	Page
GETTING HELP FROM THE SYSTEM	
Error Message Explanations	5-1
Obtaining Program Status	5-2
Program States	5-3
Obtaining System Status	5-4
WHZAT Output	5-5

INDEX	Index-1
--------------------	---------

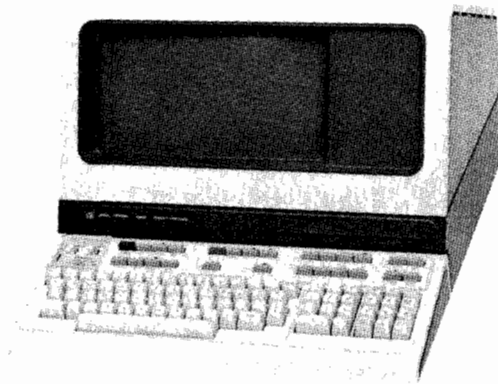
Chapter 1

Talking to the System

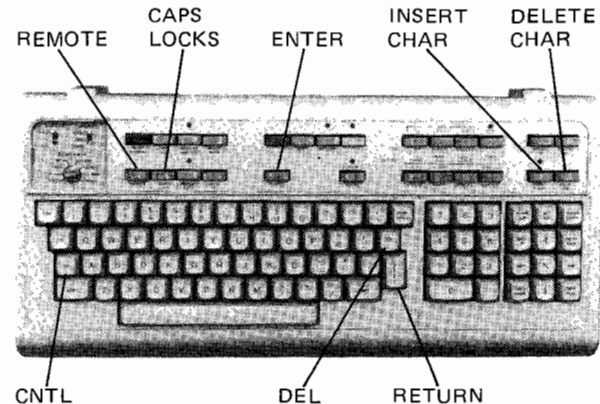
There are several ways to communicate with the HP 1000 Computer System. The most common method is through a terminal. Other methods include devices such as magnetic tape, and even another computer. Communication through these devices is merely extensions of the principles used for communication through a terminal. Your HP 1000 Model 60 or 65 System normally has one or more terminals.

The Terminal

Your terminal will probably be one of the 264X series like the one shown below.



The keyboard of this particular terminal looks like:



The keyboard is your means of communicating with the system. The system will answer you by responding on the screen of the terminal. Each character you type is displayed (echoed) on the screen.

As a result of this echoing, the displayed information that you see on the terminal consists of both your inputs and the system responses. To distinguish between the two types of information in this manual, user inputs are printed in color.

Log-On Identification

Before you can communicate with the system, you must obtain your "log-on" identification from the System Manager. The System Manager is the person responsible for the operation of your system. The log-on information is your unique means to identify yourself to the system. A sample "log-on" is:

```
CARDL.SALES<cr>
```

This entry identifies CAROL as the user who wants to communicate with the system and SALES as the group in which CAROL belongs.

Note that a carriage return is required to complete any entry into the system. The carriage return will not be shown in subsequent examples. The convention used in this manual to show variable user input is < >. For example, if a number is required as an input to the system, the sample will show <number>. Occasionally, comments will be enclosed in parentheses following the example entries.

Password

When a log-on identification is assigned to a new user, a password may also be established. To protect each user's password, the echo of it from the system is suppressed. Passwords provide a double-check of the user's identity.

Capability Level

Each user is assigned a capability level when his/her account is created by the System Manager. The account contains the log-on identification as well as other pertinent information such as what system resources can be accessed. This level indicates a set of commands the user can execute. An error will occur if an instruction requiring a higher capability level is entered. Normally, you will be assigned a level that allows you to perform all the required tasks.

This manual assumes that you have a capability level of 30. This level allows you to run most of the utility programs needed for program development. You may want to ask your System Manager what your capability level is before beginning to use this manual.

Log-On Procedure

You are now ready to "log-on" to the system.

1. Press any key on the terminal keyboard.
2. The system will respond by displaying a system prompt such as:

```
PLEASE LOG-ON: _
```

The cursor which is the blinking underscore will be after the system prompt.

3. Enter your log-on identification. The screen should look like:

PLEASE LOG-ON:CAROL.SALES

4. Check that your entry is correct. If not, you can use the BACKSPACE key to return the cursor to where the error is and retype the rest of the entry.

You may also use the DEL key (press both SHIFT AND DEL keys) to retype the whole entry.

5. Press RETURN. This indicates the following to the system:

- a. You have finished typing your message to the system.
- b. You want the system to act on what was typed.

6. If you have a password, the system will now ask for it with the query:

PASSWORD: _

Enter the password and press RETURN. You will not see the password because it is echo suppressed for your protection.

If your log-on is correct, the system will respond with a message. For example:

```
SESSION 78 ON 4:10 PM FRI., 27 MAR., 1981
PREVIOUS TOTAL SESSION TIME: 102 HRS., 26 MIN., 41 SEC.
```

A brief message from your System Manager may also appear, which you should note. A sample message is shown below.

WELCOM TO YOUR RTE-6/VM SYSTEM!

The system will be shut down for disc backup today at 12 noon. Please log-off before noon.

:

The colon is a prompt indicating that the system is ready for your next command.

The system will also notify you if there are any messages for you specifically. If there are, the following prompt will appear:

MESSAGES WAITING

:

These are messages that can be sent to you from another user or from the System Manager.

You can see what they are by entering:

:ME

If you want to delete them, enter:

:ME,,1

Error Recovery

If you make a mistake typing in the log-on identification or entering a command with several parameters, but you have not pressed the RETURN key, simply use the BACKSPACE key to go back to where the error was made and retype the remainder of the entry again. The system does not receive any characters entered until the RETURN key is pressed. You can ignore any extra characters that may appear on the screen. Once you press the BACKSPACE key, the character is erased from the system entry although it still appears on the screen.

If the error was made at the beginning of a long message and you have not pressed RETURN, it may be easier to start over. To do this, use the DEL key (pressing both the SHIFT and DEL keys) to delete the entire entry. The screen should look something like:

```
PLEASE LOG-ON:CAROL.DOOPS\  
CAROL.SALES
```

Note that the cursor returns to column 1 where you can re-enter the line.

If you have already pressed RETURN, the system will display an error message and let you try again. For example:

```
PLEASE LOG-ON:CAROL.DOOPS  
SESSION      78  LGON 04 NO SUCH USER UNABLE TO COMPLETE LOG-ON
```

If you mistype or enter a wrong password, the following message will appear:

```
PLEASE LOG-ON:CAROL.SALES  
PASSWORD?<ROSE> (The password will not  
                  be on the screen.)  
SESSION      78  LGON 05 ILLEGAL ACCESS...
```

After you log-on, if you do something that the system doesn't recognize, an error message will be displayed. Refer to chapter 5 of this manual on how to get a description of these messages with the HELP program.

Interacting With The System

Interaction with the system can be initiated by issuing system or break mode commands. Descriptions of these commands can be found in the Terminal User's Reference manual. You may not be able to use some of these commands because of capability levels assigned to these commands. The capability level needed for program development is 30 and this is the level that this manual assumes. See your System Manager if there is any question about your capability level. Some break mode commands that you may use are described below.

Break Mode

The break mode is provided for unexpected system interruptions. For example, you may want to stop a listing, a program, or a file dump from one disc cartridge to another. You can initiate the break mode by interrupting the system, e.g., pressing the carriage return and the space bar or repeating the return key. The system acknowledges the interrupt by prompting with "S=XX COMMAND?". Your session number is shown in the break-mode prompt (as a two-digit number XX). Any of the break mode commands can be entered. For example, to stop a listing in progress using the BR command:

Some useful break mode commands are:

BR	To stop the current activity such as file listing.
UP,<EQT>	To bring a device up; EQT is a number in the system equipment table that uniquely identifies the device.
RS	To abort and re-run the File Manager program.
WH	To display session status.

If you get into the break mode by accident, you can get out of it simply by pressing the RETURN key in response to the break mode prompt.

```
:LI,A92084
A92084 T=00004 IS ON CR YL   USING 00048 BLKS R=0000

0001 .NUMBER OFF;NUMBER ON 2;NUMBER ABS
0002 .NUMBER ALT 7 78
D0003 .TOF CNTL.L           (System interrupted here)
J0004 .MARGIN LEFT 5; MARGIN RIGHT 80

S=78 COMMAND ?BR           (Break mode command BR entered)
0005 A92084  DESCRIPTION  DATE CODE  PART NUMBER
0006
0007 A92085  IVB SNC       2101     92084-18999
0008 !DISK   OFLN DISC BKUP 2026     92067-16XXX
0009 !DSKUP  DISC BACKUP   2013     92067-16XXX
0010 !MTLDR  !DISK MT LOADR 2001     92067-16XXX
0011 "HELP   HELP 1        2101     92067-18XXX
0012 "HELPA  HELP 2        2101     92067-18XXX
FMGR 000
:
```

The File Manager Program

The most common means of communicating with the system is through the File Manager program. The colon is the File Manager prompt which indicates that the system is ready for a File Manager command. The system name for the File Manager program is FMGR. Usually, this program is scheduled and run every time you log-on to the system. The system makes a copy of the File Manager and names it FMGXX where XX is your session number, the same number that appears in the break-mode prompt.

The FMGR commands are described in the Terminal User's Reference Manual. Commands that you need to manage files are described in Chapter 2 of this manual.

Log-Off Procedure

When you have finished using the system, you must "log-off". To "log-off", enter:

```
:EX,SP
```

The system will respond with the system summary messages. For example:

```
FMG78 REMOVED
```

```
SESSION  78  OFF 11:58 AM  MDN., 27  APR., 1981
CONNECT TIME:           02 HRS.,  27 MIN.,  50 SEC.
CPU USAGE:              00 HRS.,  00 MIN.,  27 SEC., 930 MS.
CUMULATIVE CONNECT TIME: 209 HRS.,  31 MIN.,  44 SEC.
END OF SESSION
```

If you have any active programs, the system will first prompt for permission to terminate them.

If your response is yes, the programs are terminated and the log-off process continues as shown above. Any other response will cause the system to stop the log-off procedure. The system returns to the File Manager colon prompt.

Chapter 2

Using Disc Files

The HP 1000 system uses files to store information. Access to these files is accomplished through a program called the File Manager. The File Manager controls input to and output from disc files or peripheral devices treated as files.

Files

A file consists of sequential records. The records may be either fixed length or variable length. A file may contain ASCII (American Standard Code for Information Interchange) or binary coded information in the form of:

ASCII

- Text
- Source program
- Procedure commands
- Data

BINARY

- Compiled program
- Executable program
- Data

Files can be accessed serially or randomly and are identified by a name and the disc cartridge on which they are located. Files can be protected from other user modification or accidental erasure by assigning a security code when they are created. The file name, disc cartridge identification, and file security specification are further described below.

File Names and File Descriptors

To call for a file, or for information about a file, you use either the *file name* or the *file descriptor*.

Often, you can specify a file by its name alone, but sometimes you need to supply more than just the name. For example, file security code and disc cartridge identifier can be included as parameters after file name. The file name with parameters is called the file descriptor or namr.

Namr. The term namr is used in HP systems to specify a file. It can be a file name with parameters or a positive logical unit number. The namr parameters allow you to specify security code, file location, type of files, size of file, and size of records. The system will accept a positive number as the namr as long as the number is a logical unit number assigned to an input/output device. For example, if you use the File Manager store command (ST) to create a file from the mini-cassette, you can enter 4 as the first parameter of the ST command instead of a file name. You are using the logical unit number of the Cassette Tape Unit (4) as a file namr.



To specify a file using the namr parameters:

```
file name[:security[:cartridge[:type
[:size[:record size]]]]]
```

Example: MYFILE:HU:EK:4:10

```
file name      MYFILE
security code   HU
cartridge       EK
type of file    4  (ASCII)
size of file    10 blocks
```

If you do not care where your file is stored, with or without a security code, you can simply specify the file name after the store command. The system will store it automatically to a cartridge that you can access.

You can omit the security code, provided you include the extra colon to tell FMGR which parameter you are omitting, e.g., MYFILE::EK. You can also omit the file type, letting FMGR pick a number. Finally, record length has no meaning because these files you are creating have random-length lines, each line is stored as a record. The record size parameter is used only for fixed length records (type 2 files).

Name Syntax. When you create a file, you can use any file name you want as long as it does not already exist on the cartridge, and as long as you observe the following rules:

- Do not start a file name with a number.
- Do not use plus (+), minus (-), colon (:) or comma (.). These characters have special meaning to FMGR.
- Do not use a blank (space) embedded in the name. FMGR would use only the characters before the blank.
- Limit the name to six or less characters. FMGR ignores all after the sixth character.

The File Manager displays a message if it detects an illegal name:

```
:ST,4,9FILE
FMGR-015  (Illegal name)
:
```

Where Files are Stored

Files may be stored on many different media; the most common medium is a disc. The File Manager is used to store files on discs and other devices. The following paragraphs explain the terms associated with disc cartridges and file storage.

Disc Cartridges

Disc cartridges are logical subdivisions of a physical disc. A cartridge may directly correspond to a disc platter, or part of a platter.

Each cartridge is assigned a *cartridge reference* (CR) that distinguishes it from other cartridges on the system. Part of each cartridge is reserved by the File Manager for a *directory*. The directory contains information about each file stored on that particular cartridge.

Private, Group and System Cartridges

After a successful log-on, the system recognizes disc cartridges as either *private*, *group*, or *system* cartridges. Private cartridges can only be accessed by the assigned user and group cartridges can only be accessed by members of the group to which the cartridges are assigned. System cartridges are available to everyone in the system. However, some system cartridges contain essential system information. Access to these cartridges is limited to "read only".

Accessing Cartridges

Before any files can be created and stored on a disc cartridge, you must have a disc cartridge available. You can obtain a disc cartridge from the system disc pool by allocating one to your session.

To allocate a cartridge, enter:

```
: AC, SM
```

This assigns a cartridge to you as your private cartridge with a cartridge reference of SM. You can select any two characters as the cartridge reference. If another cartridge is already assigned the reference SM, an error message will be displayed. Simply re-enter the AC command using a different cartridge reference.

```
: AC, SM
FMGR 012    (duplicate disc label or LU)
: AC, YL
:
```

Logical Unit Number

The logical unit (LU) number is the number the system uses to access input/output (I/O) devices such as discs, magnetic tapes and printers. The operating system has a table of logical unit numbers with their corresponding physical addresses. This I/O table enables the system to find a disc cartridge, for example, when you ask for it by logical unit number. Refer to the Terminal User's Reference manual for details.

Cartridge Tape Drive

The Cartridge Tape Drive (CTD) is a high capacity, streaming magnetic tape device integrated into many of the disc drives your system may have. You can use cartridge tapes to save files from your system or to transport files between systems. Although the CTD is accessed by logical unit number, it cannot be accessed through the normal operator commands described in the RTE-6/VM Terminal User's Reference Manual that are used for other magnetic tape devices. A special utility for storing files onto the CTD is provided, the *file copy* (FC) utility. The FC utility is described in the RTE-6/VM Utility Programs Reference Manual.

Cartridge Reference Number

Whereas the disc LU number describes a specified number of tracks on whatever disc may be in a specified drive, the cartridge reference (CR) number describes a specified number of tracks on a specific disc.

Note that CR and CRN are used interchangeably; both can be either a number or two characters.

Cartridge List

When storing data, you can allow the system to put the file in the first available space on one of the disc cartridges you can access. You can also specify a disc by the cartridge identifier, either a negative LU or positive CRN number. To check the identity of the disc cartridges on your system, use the CL (cartridge list) command.

```
:CL
LU  LAST TRACK  CR  LOCK  P/G/S
33      0065      YL      P
32      0065      EK      P
.        .        .        .
.        .        .        .
```

LU is the logical unit number, LAST TRACK is the last track available on the cartridge, and CR is the cartridge reference number.

The cartridge identifier is either the LU number preceded by a minus sign or the cartridge reference (CR) number (positive).

File Directory

The File Manager maintains a directory on each disc cartridge, listing the name of each file on that cartridge, and its location. You can use the DL (directory list) command to see the list.

With no parameter, the DL command lists the directories of all cartridges you can access. However, you have the option of specifying the directory list from a particular cartridge you can access by entering either the logical unit number or the cartridge reference. If you use the LU number, make it negative to let FMGR know it is not a CR number.

```
:DL,EK
CR= EK ,-33
ILAB=DC0088 NXTR= 00048 NXSEC=064 ...
```

NAME	TYPE	SIZE/LU	OPEN TO
*EDIT	00003	00003	BLKS
LEMON	00003	00009	BLKS
EDIT1	00004	00102	BLKS
EDIT3	00004	00468	BLKS
EDTQC	00004	00048	BLKS
GLOSRY	00004	00030	BLKS
EDINDX	00004	00048	BLKS
EDIT2	00004	00310	BLKS
EDITHD	00004	00002	BLKS
LEMON	00003	00009	BLKS +001
LEMON	00003	00009	BLKS +002
LEMON	00003	00009	BLKS +003
:			

The numbers you see on the screen will be different from those shown here. The repetition of file names followed by a plus sign and a number indicates simply that those files have been increased in size. Additional disc spaces are allocated. These are called extents that are the same size as the original and are created as necessary to accommodate the increase of data.

In the captions above the list, CR is the cartridge reference number, ILAB is the cartridge label, NXTR is the next available track on the cartridge, and NXSEC is the next sector. The continuation of the line is:

```
... #SEC/TR=096 LAST TR=00059 #DR TR=01
```

The last track is number 59. However, the system uses one track for the directory, therefore the total number of tracks available to you is 58. The captions at the top of the columns in the directory list (DL command, above) give the file type and size. Memory-image program files are always type 6. ASCII-coded data is type 4. Refer to the Terminal User's Reference manual for further description of file types. The BLKS/LU entry is the number of blocks occupied by the file, where a block holds 128 16-bit words. A word holds an integer numeric value between -32768 and +32767, or two characters, numeric or alphabetic. There are 16 blocks per track.

The DL (directory list) command gives you the rest of the information you need about remaining space.

Creating Files

There are many means to create a disc file. The two most common ways are using the File Manager store command (ST) and using the interactive Editor program (EDIT). When a disc file is created, you need to furnish a name for this file. The system will assign an area on your private disc cartridge and create a directory entry for the file. You may also use the namr convention for the file to specify the needed file parameters, e.g., security code, cartridge reference, etc.

To create a file called &FILE from your terminal using the store command:

```
:ST,1,&FILE
LINE A of sample file
LINE B of sample file
Line C of SAMPLE FILE
<carriage return>
:
```

Each carriage return returns the cursor to column 1 of the next line. You will not see the colon prompt. Two carriage returns will terminate the input and the File Manager program will proceed to create the disc file specified. Note that the file name must conform to the RTE file name convention and that the file name can be entered in either upper or lower case. The File Manager will convert all lower-case letters in the file name to upper-case letters. Enter the carriage return once to start each new line. After the last line, enter carriage return twice to end the file. The File Manager will create the file with the name you entered on your private cartridge. To find out if this file exists, you can list the contents with the LIST (LI) command:

```
:LI,&FILE
&FILE T=00003 IS ON CR YL
        USING 00001 BLKS R=0000

0001 Line A of sample file
0002 Line B of sample file
0003 Line C of SAMPLE FILE
:
```

Another way to create a file is with the interactive Editor program (EDIT). This method is described in Chapter 3, Using the Editor.

Changing a File Name

You can change the name of a file you have created by using the RN (*ReName*) command while in the File Manager program (with the colon prompt). For example, rename the file &FILE to &FILE1:

```
:RN,&FILE,&FILE1
```

You can verify the new file name by listing its contents.

```
:LI,&FILE1
&FILE1 T=00004 IS DN CR YL
        USING 00002 BLKS R=0000

0001   Line A of sample file
0002   Line B of sample file
0003   Line C of SAMPLE FILE
```

If a security code was assigned to the file, it must be specified to rename the file. The security will remain in effect for the file under the new name, although the security code might be omitted in specifying the new file. For example:

```
:RN,&FILE1:36,&FILE2
:
```

The file &FILE2 will have the same security code.

Deleting Files

To remove a file from the system, use the File Manager PURGE (PU) command. To purge the file &FILE1:

```
:PU,&FILE1
:
```

Again, if there is a security assigned to the file, it must be specified.

The EDIT program provides an easy means of creating a file or modifying an existing file. To start this program:

```
:RU,EDIT
```

The Editor will respond with:

```
EDIT : Use ? for help
FI,namr specifies file to edit.
EOF
/
```

The slash (/) is the Editor prompt character and is similar to the FMGR colon (:) prompt. It indicates that the Editor is waiting for input. At this point, you may bring in an existing file for modification with the File Input (FI) command. For example, the entry FI, <oldfil> will bring in a file called oldfil for editing.

To create a new file, you must start each line with a space because the first column is reserved for Editor commands.

```
/ Line A of sample file.
/ Line B of sample file.
/ Line C of SAMPLE FILE.
```

After all lines have been entered, you must tell the Editor to create a file with a name you specified. For example, to store the above data in a file called &FILE2, enter:

```
/EC &FILE2
created file &FILE2::YL:4:2
closed file &FILE2::YL:4:2
end of edit
:
```

The EC command tells the Editor that you want to *Exit* from the Editor and *Create* a file with the specified name. The Editor will display the file namr and status, terminate, and return you to the system.

Modifying Files

File modification is accomplished through the interactive Editor. The Editor enables you to insert, delete, and replace lines or characters within a line. If your terminal has the local editing capability, the Editor allows you to use all the terminal editing features with a screen-mode operation. The following paragraphs contain descriptions of the common editing features and examples to get you started. To find out more about other capabilities of the Editor, refer to the EDIT/1000 User's Guide (P/N 92074-90001).

To modify a file, run the Editor specifying the file. For example, to modify the file just created:

```
:RU,EDIT,&FILE2  
  
EDIT : Use ? for help  
opened file &FILE2::YL:4  
3 lines read.  
Line A of sample file  
/
```

Note that the Editor reminds you to enter a ? for a display of all the Editor commands and other useful information available. You may also get a brief description of each individual command by entering a ? followed by the command.

The Editor displays the number of lines read, the first line in the file, and then prompts for a command. The line displayed is called the *pending line*. Now you may proceed to edit the file. A list of the commands that you need to start is shown below.

Command	Function
S	Get into the screen, local edit, mode.
Q	Get into local edit mode for one line only.
<RETURN>	Go to the next line.
P	Show or modify the pending line.
R	Replace pending line with new line to be entered.

-, ^ Go back one line (or enter the number of lines from the pending line, e.g., -3 to go back three lines from the pending line).

n Go to line n.

Start Screen Mode

To modify a block of lines using the terminal keyboard in the local mode, enter the screen mode as follows:

```
/S
```

Your screen will be cleared and a block of lines displayed, usually 21 lines, or to the end of file if less than 21 lines remain in the file. Each screen is bracketed by two markers. The top marker shows the line number of the first line of the screen and the bottom marker shows the last line of the screen. The top marker also shows the termination commands.

Now you can use the terminal editing and cursor positioning keys to modify the file. Note that each screen allows only 78 characters per line. Any line longer than 78 characters will cause the screen to break up the line and move the remainder of the line to a new line below.

Screen Mode Commands

The following commands are used when editing a screenful of text using the terminal edit keys and the cursor position keys.

cntl-Q Quit screen mode
 cntl-F Go to the next screen
 cntl-P Go to the previous screen
 cntl-S Start next screen at current cursor position
 cntl-X Same as cntl S but with a large screen

These commands are entered by pressing the CNTL key and the appropriate key simultaneously before pressing the return key. Each command will cause the system to read the screen and to replace the original text with your changes as shown on the screen. However, the Editor works with only a copy of your original file and until you restore the file, nothing has changed in the file. If you do not want the change made on the screen to occur, you can enter a command twice, indicated in this manual by cntl-Q cntl-Q. Doing this will cause the Editor to perform the function described above without reading the screen, leaving the file unchanged.

Once you have learned how to start the screen mode, how to move from screen to screen, and how to get out of the screen mode, you may proceed to use the cursor positioning and editing keys to modify your file. Refer to the EDIT/1000 User's Guide for other screen mode commands and features.

You may also use the Editor to edit one or more lines at a time or to edit certain characters within a line. These are the line edit and character edit modes described below.

Line Edits

You can use the line-edit mode to insert, delete or replace a complete line of text. The commands used are:

P Display the pending line
 R Replace the pending line
 I Insert a line before the pending line
 K Delete the pending line

Other useful commands are:

UN Un-do the text edit command
 -n Go back n number of lines
 +n Go forward n number of lines
 W Display a vertical window and go back to the pending line
 Ln Display n number of lines from the pending line. The default is 21 lines. The last line shown becomes the pending line.

The UN command can be used in case of a typing mistake to change the line to the original content before the modification. However, it must be entered before any other changes are made. Commands that do not change the file will not affect the UN command.

Using our sample file &FILE2, to display the pending line:

```
/P
  Line A of sample file
```

Note that all text displayed in this manner has a two column offset while text displayed in the screen mode doesn't.

To replace the pending line:

```
/RLine A is replaced with new line using command R.
/
```

To un-do a change:

```
/P                                (Display the pending line)
  Line A is replaced with new line using command R.
/Rxyxyxyxyxyxyxyxyxyxyxyxyxyxyxy
/UN
/P
  Line A is replace with new line using command R.
```

To insert a line into the file before the pending line:

```
/IThis line is inserted before line A.
/
```

To go back to the beginning of the file and display four lines:

```
/1
/L4
  This line is inserted before line A.
  Line A is replaced with new line using command R.
  Line B of sample file
  Line C of SAMPLE FILE
/
```


To delete the pending line:

```

    Line C of SAMPLE FILE
/K
EOF
/

```

When the end of the file is reached, the Editor displays EOF.

To display the file starting from line 1 with line numbers:

```

/1 W
00001 This line is inserted before line A
00002 Line A is replaced with new line using command R.
00003 Line B of sample file

EOF
/

```

Note that the W command returns to the pending line after displaying the lines. If the pending line is within the block of text displayed, a greater-than sign (>) is displayed on column 1 to point out the pending line.

More lines can be deleted if necessary. For example, if two lines are to be deleted, enter:

```

/K 2

```

This command will delete two lines starting from the pending line. You can cancel the delete command by entering the UN command immediately following the K command. Refer to the EDIT/1000 User's Guide for more line editing commands and features.

Character Edits

The character edit mode can be used to change characters in the pending line. Commands are provided in the character edit mode to insert, delete, replace, and truncate characters in the pending line. These commands are control characters used in conjunction with the P (display pending line) command and the current Editor prompt character (default is the slash).

To insert characters into the pending line, Control-S is used as follows:

```

Line B of sample file
/P//////////text <cr>
               ↑
           Control-S entered here
Line B of sample text file
/
```

The slash is used to space over characters that are not to be changed. Characters entered after the Control-S will be inserted into the pending line. Note that a space is entered after the word text before terminating the entry with a carriage return. Remember that the control characters are non-printing characters and will not be seen.

To delete characters from the pending line, use Control-C (for cancel) as follows:

```

Line B of sample text file
/P//////////-----<cr>
               ↑
           Control-C entered here
Line B of sample file
/
```

Control-C indicates the beginning of the deletion. The minus signs are place holders for the characters to be deleted. You may use any characters as place holders after Control-C. Pressing carriage return terminates the deletion.

To replace characters in the pending line, simply type the replacement as follows:

```

Line B of sample file
/P//////////text<cr>
Line B of sample text
/
```

Use the slash to move cursor to the appropriate column and enter the replacement characters. The carriage return terminates the input.

Control-T can be used to truncate the pending line:

```

/P
Line B of sample text
/P/////
               ↑
           Control-T and <cr> entered here
Line B
/
```

More than one character edit functions can be used in the same line:

```

/ This is a new line in this file.
/P////////---sample //
      ↑           ↑           ↑           ↑
      Control-S   Control-R   Control-T   Control-T entered here
      Control-C entered here
This is a sample line.
/

```

Saving the Modified File

Now that you have changed a copy of the file with the Editor program, the file with all your changes must be saved. This can be done either by replacing the old file with the modified file or creating a new file.

To create a new file and terminate the Editor, use the EC (*Exit and Create*) command and furnish a new file name. For example:

```
/EC NWFILE
created file NWFILE::YL:4:102
closed file NWFILE::YL:4:102
end of edit
:
```

The Editor displays the name of the file created (NWFILE), the cartridge where it is saved (YL), the type of file (4,ASCII), the size of the file (102 blocks), and terminates.

To replace the old file with the modified version, use the ER (*E*xit and *R*eplace) command.

```

/ER
closed file &FILE2::YL:4
end of edit
:

```

You don't have to specify the file name. However, if your file has a security code and you called up the file without specifying the security code, you must enter the security code to replace the old file. For example, if a security code of RT was assigned to &FILE2 when it was created, the security code must be specified in order to replace the file. The example below illustrates attempts to replace the file.

```

/ER                                     (1st attempt)
Illegal access &FILE2::YL:4           (File not replaced)
/ER:RT                                (2nd attempt)
opened file &FILE2:RT:YL:4
closed file &FILE2:RT:YL:4
end of edit
:                                     (File replaced)

```


Chapter 4



Program Development

The usual sequence of operations in developing a program is:

1. Create a *source* file using the Interactive Editor (EDIT).
2. Compile or assemble the source using the appropriate language processor producing a *relocatable* file.
3. Prepare the program for execution using the RTE-6/VM Loader Program (with the relocatable file).
4. Execute the *program*.

In the HP 1000 Computer System, you can compile and load in one easy step, using the CLOAD program.

There are several language processors that can be used for program development on the HP 1000 system. However, this section addresses only FORTRAN. All of the principles discussed here apply to the other language processors as well.

The source program, also called the source file, is the first file you create when you develop an application program. For this example, the source file will contain the program name, FORTRAN statements, and comments describing what the statements do.

In RTE, the source program name is conventionally five or less characters preceded by an ampersand (&) sign. The program you are about to develop is called PROG, and the source file is &PROG.

The relocatable file, also referred to as a relocatable module, contains machine code produced by the FORTRAN Compiler from the source code. The compiler produces a binary relocatable file.

If the source file name starts with an ampersand (&), and you do not specify otherwise, the compiler will name the relocatable for you by changing the ampersand to a percent sign (%). For example, %PROG will be the relocatable module derived from the source file named &PROG.

After creating the source file, use the CLOAD program to compile and load your program into the system. Following the RTE convention, the program will be given the source file name with the prefix removed. For example, an executable program called PROG will be the result of running CLOAD on &PROG.

You may also comopile and load a program separately using the appropriate compiler or assembler and the Relocating Loader (LOADR).

If the relocatable name starts with a percent sign, and you do not specify otherwise, the Loader will name the program by removing the percent sign. For example, %PROG will result in an executable program called PROG.

The program you are about to write is short and simple. When you run it, it will ask you for the length of a pendulum and show the oscillation of this pendulum.

Writing the Source Program

Run the Editor and start writing PROG:

```
:RU,EDIT

/ FTN,L
/      PROGRAM PROG(3,89)
/ C
/ C THIS PROGRAM COMPUTES THE TIME OF
/ C OSCILLATION OF A SIMPLE PENDULUM.
/      LU=LOGLU(SYLU)
/      CALL FSYSU (1,1)
/      WRITE (LU,10)
/      10 FORMAT ('LENGTH OF PENDULUM (REAL # IN INCHES)
/      READ (LU,*) R
/ C
/ C      INITIALIZE CONSTANTS
/ C
/      PI=3.14159
/      G=32.*12.
/ C COMPUTE OSCILLATION TIME
/      T=PI*(R/G)**.5
/ C DISPLAY ANSWER
/      WRITE(1,33) T
/      33 FORMAT (1X,"TIME=",F6.2," SECONDS")
/      END
```

Remember to space once at the start of each line. You can use the TAB key on your terminal to start your program statements on column 7. Tabulation can be reset for other languages. Refer to the Setting Tabs section in Chapter 2 of the EDIT/1000 User's Guide.

If you make an error, use the EDIT commands given in the previous chapter to make the correction.

The first line of the program is a compiler control statement. FTN,L specifies a source listing.

Line 2, the PROGRAM statement, includes program name, type, and priority (3,89). You can default these values by using just the empty parentheses (), followed by a comma to delimit the comment that follows on the same line. If there is no comment, the parentheses and the comma are not required. The compiler and loader listing is shown below.

In line 6, LOGLU is a system subroutine that gets the logical unit number of the terminal from which the program is being run. The Loader will find LOGLU in one of the libraries on the system disc.

When you have the source file completed, use the EC (Exit and Create) command to exit the Editor, saving the file under the name &PROG:

```
/EC,&PROG
```

Compile and Load a Program

You can compile and load the program in one step. This can be done with the CLOAD (Compile and LOAD) program. After you have created (or corrected) the source program, run CLOAD with the following parameters:

```
source file name
list device (optional)
relocatable file name (optional)
control statement (optional)
```

The optional control statement allows you to override the control statement in the source file if needed. See the following example showing how to compile and load a program. The displayed information other than the program itself may be different on your system.

:CLOAD,&PROG

PAGE 1 FTN. OPTS: LYI 3:27 PM WED., 26 AUG., 1981

```
2      PROGRAM PROG
3 C
4 C THIS PROGRAM COMPUTES THE TIME OF
5 C OSCILLATION OF A SIMPLE PENDULUM.
6      LU=LOGLU(SYLU)
7      CALL FSYSU (1,1)
8      WRITE (LU,10)
9      10 FORMAT ('LENGTH OF PENDULUM (REAL # IN INCHES)')
10     READ (LU,*) R
11 C
12 C INITIALIZE CONSTANTS
13 C
14     PI=3.14159
15     G=32.*12.
16 C COMPUTE OSCILLATION TIME
17     T=PI*(R/G)**.5
18 C DISPLAY ANSWER
19     WRITE(1,33) T
20     33 FORMAT (1X,"TIME=",F6.2," SECONDS")
21     END
```

FTN4X COMPILER: HP92834 REV.2101 (801125)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 117 COMMON: (NONE)


```
$END FTN4X:  NO DISASTERS,  NO ERRORS,  NO WARNINGS.
PROG      36042 36226
```

```
LOGLU  36227 36304  92067-1X297 REV.2013 790228
FSYSU  36305 36316  24998-1X323 REV.2101 800731
.FIOI  36317 36376  24998-1X322 REV.2101 800803
ER0.E  36377 36377  24998-1X249 REV.2001 750701
.EIO.  36400 37624  24998-1X329 REV.2101 801107
FMTIO  37625 41056  24998-1X328 REV.2101 800929
.FMCV  41057 43321  24998-1X333 REV.2101 800709
.FIO.  43322 43576  24998-1X330 REV.2101 800708
.IDER  43577 43712  24998-1X321 REV.2101 800731
.UFMP  43713 43725  24998-1X296 REV.2101 800731
REID   43726 44052  92067-1X275 REV.2013 790316
.EXIT  44053 44126  24998-1X320 REV.2101 800731
PAU.E  44127 44127  24998-1X254 REV.2001 750701
PNAME  44130 44200  92068-1X035 REV.2101 800919
.IDCL  44201 44302  24998-1X305 REV.2101 800731
.IDCM  44303 44346  24998-1X327 REV.2101 801007
ERR0   44347 44436  24998-1X250 REV.2001 771122
SQRT   44437 44540  24998-1X181 REV.2001 780424
LIMEM  44541 44561  92067-1X477 REV.2013 790126
.OPN?  44562 44605  24998-1X325 REV.2101 800803
```

```
  5 PAGES RELOCATED      5 PAGES REQ'D      NO PAGES EMA  NO PAGES MSEG
LINKS:BP  PROGRAM:BG  LOAD:TE  COMMON:NC
/LOADR:PROG  READY AT  3:28 PM  WED., 26  AUG., 1981
/LOADR:$END
```

```
/CLOAD: END
```

```
:
```

Now you can run your program under the name PROG.

Implied Run

On the Model 60 or 65 system, you can use the implied run instead of the RU command. For example, try:

```
:PROG
LENGTH OF PENDULUM (REAL # IN INCHES)
22
TIME= .75 SECONDS
:
```

If there is no such program as the one you entered, the File Manager will display an error message. For example, try the implied run on a non-program file:

```
:SWAP
FMGR 067
:??
FMGR 067 PROGRAM NOT FOUND
:
```

Saving a Program

The term saving a program relates to the fact that FMGR removes from memory the program you just compiled and loaded each time you log-off from the system. This means the next time you log-on to the system, you no longer can run that program. To save the program so that you can run it in subsequent sessions, use the File Manager SP command before you leave the current session.

```
:SP,PROG
:
```

Chapter 5

Getting Help From the System

When you have initiated some action and the results are not what you expected or the action you requested is not taking place, you can obtain help from the system. This can be in the form of an error message explanation, the status of a program, or the status of the system.

Any unsuccessful action usually will cause an error message to be displayed at your terminal. You can obtain a brief explanation of the message.

If you get into an unfamiliar situation, you can usually get into the break mode by pressing any key. The break-mode prompt will be displayed on your screen. Once in the break mode, you can obtain the status of a program or the system by means of the WHZAT program. This program is described later in this chapter.

Error Message Explanations

When the File Manager program cannot interpret your input or that input has caused a recognized problem, an error message is displayed in the form:

FMGR nnn or FMGR-nnn

where nnn is a three-digit number. Note that there are positive and negative error numbers.

A brief description of this error can be obtained by entering:

:??

Example:

```
FMGR-006
:??
FMGR-006 FILE NOT FOUND
:
```

This message indicates that the requested file could not be found. You could also have requested an explanation of an error code by providing the number:

```
:??,15
FMGR 015 LS TRACK REPORT

:??,-15
FMGR -15 ILLEGAL NAME
```

If you need further explanation of an error, you can use the HE (help) command. This command will provide further explanation of the error just occurred.

```
FMGR-006
:??
FMGR -06 FILE NOT FOUND.
:HE
FMGR-006
FILE NOT FOUND
AN ATTEMPT WAS MADE TO ACCESS A FILE
THAT CANNOT BE FOUND. CHECK THE FILE
NAME OR THE CARTRIDGE REFERENCE.
```

:

Obtaining Program Status

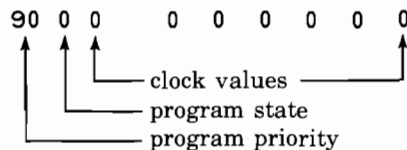
There are times when you may think that the system has ignored your request. You can find out what the status of your program is by using the SStatus command.

The ST command is used when the system is currently processing a request and there is no colon prompt. When you press any key, the system is interrupted and enters into the break mode by displaying a prompt. A Sample break-mode prompt is:

```
S = 55 COMMAND ?
```

Usually, the number will be the session number of your terminal. After receiving the prompt, entering the ST command followed by the program name will display a line of information about that program. For example:

```
S = 55 COMMAND ?ST,TEST
```



The first number is the priority assigned to the program. This is an integer value in the range of 1 to 32767, with 1 being the highest in the priority order.

The second number indicates the current state of the program. In the example, the program is in the 0 (dormant) state. Other states of interest to you are:

PROGRAM STATE

ACTION

- | | |
|-----------------|---|
| 1 (Scheduled) | The program is scheduled and may be executing or waiting to be executed. |
| 2 (I/O Suspend) | The program is suspended and waiting for an Input/Output operation to complete. |

3 (General Wait)	The program is suspended waiting for system resource such as a printer or tape drive.
4 (Unavailable Memory Suspend)	The program is suspended and waiting for the required memory to become available.
5 (Disc Allocation Suspend)	The program is suspended because it cannot get the required system disc tracks.
6 (Operator or Program Suspend)	The program has been suspended by an operator command or another program.
9 (Background Segment)	The program is a segment and has no status independent of the main program to which it belongs.

The program states section below summarizes the action you may take if your program is in a particular state.

The remaining numbers displayed by the ST command are usually all zeros unless the program is scheduled by the system clock. In this case, the numbers indicate the next time the program is to be scheduled for execution.

Program States

PROGRAM STATE	ACTION
0 Dormant	Schedule the program again if you want to run it.
1 Scheduled	Terminate the program if you are the person who scheduled it. Try it again, making sure that all parameters are correct and all devices to be used are operating and ready.
2 I/O Suspend	Check the I/O device in question. Make sure that it is ready and available. You may have to wait for another program to finish using the device before your program can use it.
3 General Wait	Either continue to wait or make the resource available to the waiting program.
4 Unavailable Memory Suspend	Consult the system manager about getting more memory.

- 5 Disc Allocation Suspend Check with the system manager for the reason why there are no system tracks available. You can try to release tracks from your own programs with the RT (release tracks) command. Enter RT,<program name>.
- 6 Operator or Program Suspend Try typing the GO command to continue execution.

It can be used to display:

- All scheduled and suspended programs and their status.
- The status of all partitions in numeric sequence.
- Only those programs associated with the user session (default mode).

To run WHZAT:

```
:RU,WHZAT
```

It can also be run while in the system break mode:

```
S = xx COMMAND ?WH
```

Obtaining System Status

The current system status is obtained by executing the system program called WHZAT. WHZAT is a utility program that provides current system environment information.

The information displayed is preceded by a heading which includes the current system time and column headings. A sample printout of the WHZAT program is shown below.

```
:WH
13:14:14:100
-----
PRGRM T PRIOR PT  SZ DO.SC.IO.WT.ME.DS.OP.      .PRG CNTR.  .NEXT TIME.
-----
**FMG30 3 00090 20  12 * * * *  3,WHZAT * * * * * P:50173
  WHZAT 1 00041  0   . . 1,  . . . . . P:73136
.
-----
ALL LU'S OK
ALL EQT'S OK
-----
13:14:14:960
:
```

There are other options that display various system information, e.g., partition status, all program status, etc. To see the status of all the scheduled and suspended programs, the AL option is used. To display the status of all active programs in the system, the PR option is used. Refer to the RTE-6/VM Utility Programs Reference Manual for further description of the WHZAT program.

WHZAT Output



When you run WHZAT, your program may be listed in one of the following states:

- 0 dormant
- 1 scheduled
- 2 I/O suspended
- 3 suspended in general wait list
- 4 suspended waiting for system available memory
- 5 suspended waiting for disc tracks
- 6 either operator or program suspended

In the case of the AL option, the WHZAT output consists of three distinct blocks. The first block shows the user session programs. The second block shows state 3 programs having father-son relationship. The calling (father) programs are identified with two stars. The third block consists of all other individual programs.

All locked LUs and EQTs are displayed at the end as follows:

```
LOCKED LU'S (PROG NAME) xx(program name),--,xx(program name)
LOCKED EQT'S (PROG NAME) xx(program name),--,xx(program name)
```

If no LU or EQT is locked, the corresponding message is suppressed. Also displayed at the end is the maximum contiguous free tracks available at that instant of time. The corresponding LU number is also displayed. For example:

```
LOCKED LU'S (PROG NAME)      6(SPOUT),
MAX CONT. FREE TRKS :      6, LU 3
```

The following information is displayed for state-4 programs (memory suspended).

```
MAX CONT. SAM AVAIL:      xxx
TOTAL SAM AVAILABLE:      xxx
MAX CONT. SAM EVER AVAIL:  xxx
```

The program counter value is displayed for every program irrespective of the state. The message P:xxxxx is displayed under the PRG CNTR column. If a program is swapped out, SWP will appear at the end of the octal number (i.e., P:xxxxxSWP).

If a program is in state 2 (I/O suspended) and the EQT is down, the following message appears:

```
2,EQ:xxx,DN,AV:x,ST:xxx
```

where:

xxx is the EQT number in decimal

AV:x is the availability code (x which is driver independent)

0 available

1 disabled

2 busy

3 waiting for DMA

ST:xxx is the status (xxx in octal) of the EQT in operation

However, if a LU is down, the message is:

```
2,LU:xxx DN,EQ:xx,AV:x,ST:xxx
```


In a real-time situation, the following condition might occur. PRGX invokes PRGY and suspends itself until PRGY terminates. PRGY in the meantime invokes PRGX, thus a deadly embrace (DEADLOCK) condition is created. A program scheduled by another program is considered as having a father-son relationship; In the deadlock case, the following message would appear:

```

PRGX          *      *      3,PRGY
PRGY          .      .      3,PRGX'S QUEUE
** BLOCK **
**          * DEADLOCK **
**          * SEE ABOVE FOR REPORT ON PRGX **

```

As WHZAT executes, the state of the system changes constantly. A case might occur in which the status of a program is reported and in the meantime that program was invoked by another program. The status of that might be reported again as the son of the calling program. For example:

```

PRGA          *      *      3,PRGX
PRGX          .      1

PRGB          **          3,PRGX
**          * SEE ABOVE FOR REPORT ON PRGX **

```

When a program is in state 3 (general wait state), WHZAT will display a message giving the reason for the wait. These messages and the reasons are listed in Table 5-1. The SM program status mode is similar to the AL mode output.

Table 5-1. General Wait State Messages

MESSAGE	REASON FOR WAIT
LULK lu,LKPRG=progx	The listed program attempted to put a lock on logical unit LU. Program progx already has a lock on lu. The listed program will be rescheduled when progx removes its lock.
RN xx,LKPRG=progx	The listed program attempted to set resource number xx. Program progx already has a lock on the resource number. The listed program will be re-scheduled when progx removes the lock.
RESOURCE	The listed program attempted to allocate a resource number. The system has no more resource numbers available. The operating system will re-schedule the listed program when a resource number is available.
CLASS #	The listed program requested a class number but the system has no more available. The operating system will reschedule the listed program when a class number becomes available.
CL xx	The listed program is waiting on completion of a class GET to class number xx.
progx	The listed program scheduled progx with wait. The listed program will be rescheduled when progx completes.
progx'S QUEUE	The listed program scheduled progx on the queue with wait. progx is not dormant so the listed program must wait. The listed program will be re-scheduled after the scheduling of progx completes.
BL,EQT xx	Buffer limit exceeded on the controller in EQT entry xx.
EQLK xxx,LKPRG= PRGA	Program suspended for a locked EQT.
EQLK TABLE FULL	Program attempts to lock an EQT and the EQT table is full.

Table 5-2. WHZAT Display Abbreviations

PROGRAM MODE	
PRGRM	Program name.
T	Program type.
PRIOR	Program priority.
PT	Partition number; 0 means memory-resident.
SZ	Page size of program; ** means memory-resident.
DO	Dormant (state 0).
SC	Scheduled (state 1).
IO	I/O suspended (state 2).
WT	General wait state (state 3).
ME	Memory suspended (state 4).
DS	Disc suspended (state 5).
OP	Operator suspended (state 6).
PRG CNTR	Point of suspension; number shown in octal.
NEXT TIME	Time program is listed on time list.
A	After the partition number means the program was assigned to the partition.
E	After the program's type means it is an EMA program.
B	After the program's priority means the program is running under batch.

Table 5-2. WHZAT Display Abbreviations (Continued)

PARTITION MODE	
M	Mother partition.
C	Subpartition in chain mode.
S	Subpartition available.
R	Reserved partition.
PTN#	Partition number.
SIZE	Page size of program.
PAGES	Physical pages where program resides.
BG	Background program.
RT	Real-time program.
PRGRM	Program name.
SHR	Shareable EMA partition (see note 1 below).
LBL	Label of shareable partition.
#ACT	Number of current users of shareable EMA partition.
L	Partition lock status.
<p>NOTE 1: This is indicated by "SH" in the field. Other possible entries and their messages are:</p> <p>** Partition is a mother partition with a subpartition that is shareable.</p> <p>* Partition is a subpartition of a shareable mother partition.</p>	

Table 5-2. WHZAT Display Abbreviations (Continued)

PROGRAM LIST MODE	
NAME	Program name.
TY	Program type.
PRIOR	Program priority.
LMAIN	Low memory address.
HMAIN	High memory address.
LOBP	Low base page.
HIBP	High base page.
SZ	Program size (pages).
EMA	EMA size.
MSEG	Number of pages in MSEG.
LBL	Label of shareable EMA partition if program uses it.
PTN	Partition number if the program is assigned to one.
TM	Temporary (TE) or permanent (PE). Blank if memory-resident.
COM	System common type (SC,RC,NC).
LU	Logical unit program is stored on.
S-ID	Session identifier if the program is loaded under Session Monitor.



Index

A

accessing cartridges, 2-3

B

BLKS/LU, 2-5

block, 2-5

break mode, 1-5

C

capability level, 1-2

cartridge

 identifier, 2-4

 label, 2-5

 list, 2-4

 reference number, 2-3

 tape drive, 2-3

changing a file name, 2-6

character editing, 3-6

CL (cartridge list), 2-4

compile and load a program, 4-3

CR (cartridge reference number), 2-3

creating files, 2-5

CTD, 2-3

D

deleting files, 2-6

disc cartridges, 2-2

DL (directory list), 2-4

E

error

 message explanations, 5-1

 recovery, 1-4

F

file

 descriptor, 2-1

 extents, 2-4

 name, 2-1

 name syntax, 2-2

 types, 2-5

File Manager program, 1-6

Index

G

getting help from the system, 5-1
group cartridge, 2-3

I

I/O (input/output) devices, 2-3
implied run, 4-6
interacting with the system, 1-4

L

label, cartridge, 2-5
line editing, 3-3
log-off procedure, 1-6
log-on
 identification, 1-2
 procedure, 1-2
logical unit number, 2-3
LU (logical unit), 2-3

M

modifying files, 3-1

N

namr
 description, 2-1
 parameters, 2-1

Index-2

O

obtaining
 program status, 5-2
 system status, 5-4

P

password, 1-2
private cartridge, 2-3
program
 development, 4-1
 states, 5-3

S

saving
 a program, 4-6
 modified file, 3-7
screen
 mode, 3-2
 mode commands, 3-2
source program, 4-1
system cartridge, 2-3

T

terminal keyboard, 1-1
Track, 2-5

U

using files, 2-1

W

where files are stored, 2-2

WHZAT

output, 5-5

program, 5-4

Word, 2-5

writing the source program, 4-2



**HEWLETT
PACKARD**

**MANUAL PART NO. 92084-90002
Printed in U.S.A. December 1981**

**HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014**