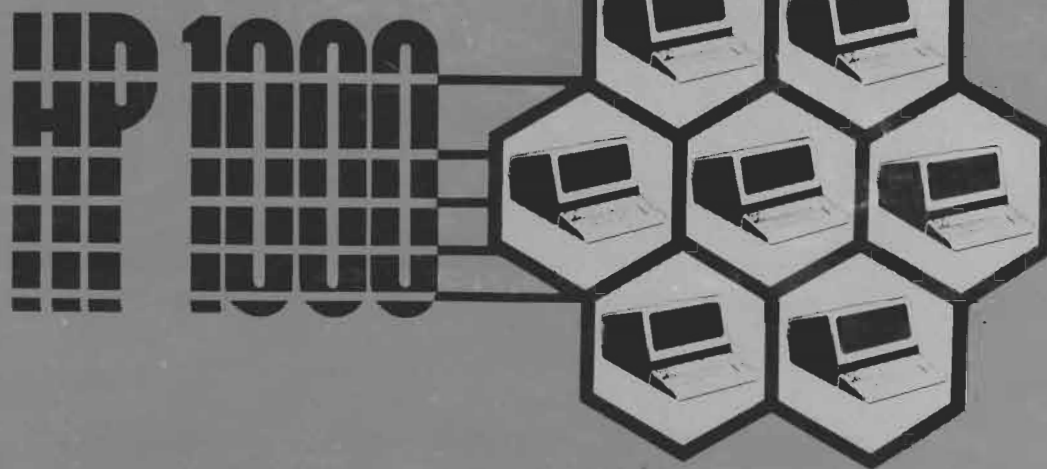


Getting Acquainted
With

RTE-IVB



 **HEWLETT
PACKARD**

PREFACE

This manual is an introduction to the RTE-IVB Operating System (HP 92068A) with Session Monitor. It illustrates the ease with which you can do the following:

- Communicate with the System
- Create and Modify Files
- Develop and Run Programs
- Execute system commands (or run programs) from a procedure file

This manual should be used as a tutorial guide for the first-time user. Try each example as you read the sections. Shown here is only a small subset of the capabilities and commands that are available with RTE-IVB. For more detailed information, refer to the RTE-IVB Terminal User's Reference Manual and/or the RTE-IVB Programmer's Reference Manual.

HP Computer Museum

www.hpmuseum.net

For research and education purposes only.



CONTENTS

| | |
|-----------------------------------------------|------|
| Section I | Page |
| TALKING TO THE SYSTEM | |
| The Terminal | 1-1 |
| Point-To-Point and Multipoint Terminals | 1-2 |
| Special Keys | 1-2 |
| Log-On Identification | 1-3 |
| Password | 1-3 |
| Capability Level | 1-3 |
| Log-On Procedure | 1-3 |
| Error Recovery | 1-4 |
| Interacting with the System | 1-6 |
| Log-Off Procedure | 1-7 |
| Command Summary | 1-7 |
| Log-On Procedure | 1-7 |
| Error Recovery | 1-8 |
| System Commands | 1-8 |
| Log-Off Procedure | 1-8 |
| Example Summary | 1-9 |
| Section II | Page |
| USING FILES | |
| Files | 2-1 |
| Where Files are Stored | 2-2 |

| | |
|-------------------------------------------|------|
| Disc Cartridges | 2-2 |
| Private Group and System Cartridges | 2-2 |
| Accessing Cartridges | 2-3 |
| Accessing Devices | 2-3 |
| Creating Files | 2-4 |
| Modifying Files | 2-7 |
| Line Edits | 2-7 |
| Character Edits | 2-10 |
| Multipoint Terminal Character Edits | 2-12 |
| Saving the Modified File | 2-15 |
| Changing a File Name | 2-16 |
| Deleting Files | 2-17 |
| Command Summary | 2-17 |
| Example Summary | 2-22 |

| | |
|----------------------------------|------|
| Section III | Page |
| DEVELOPING A PROGRAM | |
| Source Program Reminders | 3-1 |
| FORTRAN Program Statements | 3-1 |
| End Statement | 3-1 |
| File Name Convention | 3-2 |
| Writing a Source Program | 3-2 |
| Compiling a Source Program | 3-2 |
| Loading a Program | 3-4 |

CONTENTS (CONT.)

| | |
|----------------------------------|------|
| Executing a Program | 3-6 |
| Program Error Handling | 3-6 |
| Compile and Load a Program | 3-6 |
| Command Summary | 3-6 |
| Basic Programs | 3-8 |
| Example Summary | 3-11 |

| | |
|---------------------------------|------|
| Section IV | Page |
| USING PROCEDURE FILES | |
| Creating a Procedure File | 4-1 |

| | |
|----------------------------------|-----|
| Executing a Procedure File | 4-1 |
| Command Summary | 4-6 |
| Example Summary | 4-6 |

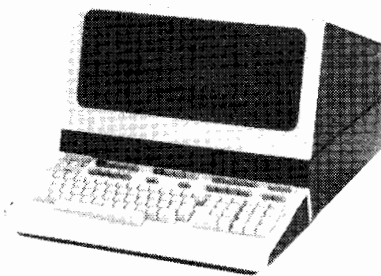
| | |
|-------------------------------------|------|
| Section V | Page |
| GETTING HELP FROM THE SYSTEM | |
| Error Message Explanations | 5-1 |
| Obtaining Program Status | 5-2 |
| Program States | 5-4 |
| Obtaining System Status | 5-4 |

Talking to the System

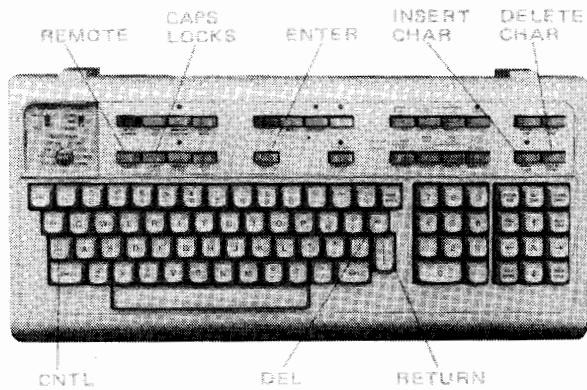
There are several ways to communicate with the HP 1000 Computer System. The most common method of communication is through a terminal. Other methods include devices such as magnetic tape, and even another computer. Communicating through these devices are merely extensions of the principles used for communication through a terminal.

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. So that you can keep track of what you are typing, the system returns (echos) each character typed by displaying it on the terminal screen.

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

Point-To-Point and Multipoint Terminals

Before using the terminal to communicate with the system, it is necessary to determine how the terminal is connected to the system.

If your terminal requires a telephone connection, obtain the telephone number from the System Manager and do the following:

1. Dial the computer telephone number.
2. Wait for the computer to return its "carrier signal". For acoustically coupled telephone equipment this signal will be audible as a high pitched tone in the phone handset; for other equipment a CARRIER light will go on. If you do not get the carrier signal, the phone line may be busy (busy signal heard) or the computer may not be in operation at this time.
3. When the carrier signal is received, place the handset firmly in the receptacle (or, for some equipment, press the DATA button).

Refer to the illustration of the keyboard and locate the keys marked REMOTE and CAPS LOCK. These two keys must be in the down or depressed position for communication with the system. Now, press any key and note the results on the terminal screen. If the message

PLEASE LOG-ON:

1-2

appears, the terminal is a POINT-TO-POINT terminal. If this message does not appear after a few seconds, press the key marked ENTER. If the above message appears now, the terminal is a MULTIPOINT terminal and the ENTER key must be used instead of the RETURN key to terminate the messages you are sending to the system. If this message still does not appear, contact the System Manager.

NOTE

If "*" appears instead of the above message, you are using the system console. Press RETURN and move to another terminal to try again.

Special Keys

Some of the special keys you may be using are labeled DEL (delete), CNTL (control), INSERT CHAR (insert character), and DELETE CHAR (delete character).

If you are using a point-to-point terminal, you will be concerned with DEL and CNTL. The key labeled DEL (upper case) is used to completely delete the message you are sending to the system. The key labeled CNTL is used in conjunction with other keys to indicate special functions to the system. You will see examples of how these keys are used later on.

If you are using a multipoint terminal, the DEL and CNTL keys on your terminal will not be used. Instead, you will use the INSERT CHAR, DELETE CHAR, and other keys to perform the same functions. These differences will be pointed out as their uses are described.

LOG-ON IDENTIFICATION

Before you can communicate with the system, you must obtain your "log-on" identification (your means to identify yourself to the system) from the System Manager. A sample "log-on" identification is:

CAROL.MANUF

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

Password

When an account is assigned, a password may also be assigned to it. 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 account is created. This level indicates which set of commands the user can execute. An error will occur if an instruction with a higher capability level is entered.

This manual assumes that the user has a capability level of at least 30. This level allows the user to control programs. You may want to ask the System Manager what your level is before beginning to use this manual.

LOG-ON PROCEDURE

You are now ready to "log-on" to the system. Remember that you must use ENTER instead of RETURN if you are using a multipoint terminal.

To log-on to your system:

1. Press any key (the ENTER key if using a multipoint terminal).

2. The system will respond by displaying:

PLEASE LOG-ON:

3. Respond to the system message by typing your "log-on" identification. The screen should now look similar to:

PLEASE LOG-ON: CAROL.MANUF

4. Press RETURN. Pressing the RETURN key 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.

5. 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 appear since it is echo suppressed.

If your log-on is correct, the system will respond with a welcome message similar to:

```
SESSION    9    ON    1:51 PM THU., 24 MAY , 1979
PREVIOUS TOTAL SESSION TIME:4602 HRS., 57 MIN., 51 SEC.
```

A brief message from the system may also appear, which you should note. For example:

WELCOME TO YOUR RTE-IV SYSTEM !!!!!

The system will also notify you if there are any messages waiting for you in your message file with:

MESSAGES WAITING
:

These are messages that can be sent to you from other users or from the System Manager.

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

1-4

Error Recovery

If a mistake is made typing in your "log-on" identification or password, but you have not pressed RETURN, simply use the key labeled BACKSPACE to go back to where the error was made and retype the remainder of the message again.

If the error is 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 keys labeled SHIFT and DEL to delete the entire message. The screen should look something like:

```
PLEASE LOG-ON:  CAROL,MANUF\
CAROL,MANUF
```

when you have reentered the line.

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

For example, the "log-on"

```
PLEASE LOG-ON: CAROL,MANUF  
SESSION      9  LGON 04 NO SUCH USER  
              UNABLE TO COMPLETE LOG-ON
```

would provide this message which was caused by typing a comma (,) instead of a period (.).

Another error may be displayed if you entered your "log-on" identification correctly, but mistyped or entered a wrong password. For example:

```
PLEASE LOG-ON: CAROL.MANUF  
PASSWORD? [REDACTED]  
SESSION      9  LGON 05 ILLEGAL ACCESS  
              UNABLE TO COMPLETE LOG-ON
```

If this occurs, the "log-on" process must be repeated.

There is another message that may appear. This is:

```
PLEASE LOG-ON: CAROL.MANUF  
SESSION      9  LGON 03 SESSION LIMIT EXCEEDED  
              UNABLE TO COMPLETE LOG-ON
```

This error is not caused by a typing mistake. It states that the number of users allowed to "log-on" the system at the same time has reached its limit. If you receive this message, you must wait until someone "logs-off" before you may "log-on".

INTERACTING WITH THE SYSTEM

Interaction with the system is initiated by issuing commands to the system to perform the desired functions. The system indicates that it is waiting for a command by displaying a colon (:) prompt.

Issue a command to the system by entering:

:SYTI

after the prompt. This command requests the system to display the current year, Julian date (the number of days elapsed since January 1 plus one), and the time of day. The system response should look like:

| | | | | |
|------|----------------|-------|---------|---------|
| 1979 | 002 | 12 | 33 | 15 |
| ↑ | ↑ | ↑ | ↑ | ↑ |
| year | | hours | | seconds |
| | Julian date | | minutes | |

This time is kept by the clock within the computer.

If the message:

MESSAGES WAITING

was displayed when you logged onto the system, entering the command:

:ME

will display these messages to your terminal. For example:

:ME

Description of "from whom & when the msg. was sent".

MANAGER.SYS 1:30 AM MON. 1 JAN., 1979
WELCOME FROM YOUR SYSTEM STAFF

LOG-OFF PROCEDURE

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

:EXIT,RP (or **EXIT,SP**)

The system will prompt for permission to terminate any active program, log you off by dismounting all private cartridges mounted, terminating any active programs, updating your accounting information and issuing a final message as follows:

```
FMG09 REMOVED
SESSION    9   OFF  2:00 PM THU., 24 MAY , 1979
CONNECT TIME:          00 HRS.,  00 MIN.,  05 SEC.
CPU USAGE:           00 HRS.,  00 MIN.,  00 SEC.,   240 MS.
CUMULATIVE CONNECT TIME: 4602 HRS.,  59 MIN.,  19 SEC.
```

Enter EXIT,RP if you no longer need the contents of your disc cartridge. This will release the disc cartridge for other users. The contents will be erased when the disc is accessed by another user with the AC command. If you want to keep the contents of your disc cartridge, use EXIT,SP.

COMMAND SUMMARY

Log-On Procedure:

1. If necessary, dial up the computer over telephone lines.
2. Press any key (Remember: Press ENTER for multipoint terminals).

3. Wait for log-on message.
4. Type "log-on" identification.
5. Press RETURN or ENTER.
6. If you have a password; wait for password message, type password, and press RETURN or ENTER.
7. Wait for prompt (:).

Error Recovery:

1. To correct a character, use BACKSPACE and retype the character.
2. To delete a command, use SHIFT and DEL.

System Commands:

1. Display system time

:SYTI

2. Display messages

:ME

Log-Off Procedure:

:EXIT,RP or
:EXIT,SP

EXAMPLE SUMMARY

PLEASE LOG-ON: CAROL.MANUF

PASSWORD?

SESSION 9 ON 1.51 PM THU., 24 MAY, 1979
PREVIOUS TOTAL SESSION TIME: 4602 HRS., 57 MIN., 51 SEC.
WELCOME TO YOUR RTE-IVB SYSTEM !!!
MESSAGES WAITING

:SYTI

1979 001 12 33 30

:ME

MANAGER.SYS 1:30 AM MON., 1 JAN., 1979
WELCOME FROM THE SYSTEM STAFF

:EXIT,RP

FMG09 REMOVED

SESSION 9 OFF 2:00 PM THU., 24 MAY, 1979
CONNECT TIME: 00 HRS., 00 MIN., 05 SEC.
CPU USAGE: 00 HRS., 00 MIN., 00 SEC., 240 MS.
CUMULATIVE CONNECT TIME: 4602 HRS., 59 MIN., 19 SEC.

Using 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.

| | | | | | |
|---------------------------|---------------------------|---------------------------|------------|---------------------------|-------------|
| Record 1 | Record 2 | Record 3 | ... | Record n | FILE |
|---------------------------|---------------------------|---------------------------|------------|---------------------------|-------------|

The records may be either FIXED LENGTH or VARIABLE LENGTH.

A file may contain ASCII (American Standard Code for Information Interchange) coded information in the form of:

- Text
- Source programs
- Procedure commands
- Data

or BINARY coded information in the form of:

- 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. The file name consists of 1 to 6 ASCII characters that must conform to the following restrictions:

- Only printable characters can be used.
- Plus (+), minus (–), colon (:), or comma (,) are not allowed
- The first character must not be blank (space) or a number (0-9)
- Embedded blanks are not allowed

and several optional subparameters that specify the disc and security code.

Files can be protected from other user modification by assigning a security code when the file is created. A positive security code (a number greater than zero or two ASCII characters) restricts other users from writing into the file, but not from reading it. That is, a user who does not know the code may read the file, but may not write into the file. A negative security code (a number less than zero) restricts all access to the file unless the security code is specified. A zero security code implies no protection.

2-2

WHERE FILES ARE STORED

Files may be stored on many different mediums; the most common being a disc. The File Manager is used to store files on discs and other devices.

Disc Cartridges

DISC CARTRIDGES are logical subdivisions of a physical disc. A cartridge may directly correspond to a disc platter, or may be defined as part of a platter.

Each cartridge is assigned a CARTRIDGE REFERENCE that identifies 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 that is stored on that particular cartridge.

Private, Group and System Cartridges

The system recognizes disc cartridges as either PRIVATE, GROUP or SYSTEM cartridges. PRIVATE cartridges can only be accessed by the assigned individual user and GROUP cartridges can only be accessed by the members of the group to which the cartridges are assigned. SYSTEM cartridges (defined as logical unit 2 and 3) contain the system information. Users other than the System Manager have "read only" access to System cartridges. SYSTEM GLOBAL cartridges (all other system cartridges) can be accessed by any user.

Accessing Cartridges

Before any files can be created and stored on a disc cartridge, you must first allocate a disc cartridge. Mounting a cartridge allocates the disc space and creates an entry for the cartridge in the system cartridge directory. This is accomplished by using the AC command.

To allocate a cartridge, enter the command:

```
:AC,25
```

This mounts a cartridge to you as PRIVATE and assigns a cartridge reference of 25. If cartridge 25 is already mounted, an error message will be displayed. Simply re-enter the AC command using a different cartridge reference.

```
:AC,25
FMGR 012 (Duplicate disc label or LU)
:AC,26
:
```

To find out what cartridges are mounted for use, enter the command:

```
:CL
```

A list of private, group and system cartridges will appear in the form:

| LU | LAST TRACK | CR | LOCK | P/G/S |
|----|------------|-------|------|-------|
| 19 | 0202 | 00026 | | P |
| 18 | 0202 | 00025 | | G |
| 02 | 0175 | 00002 | | S |

↑
Logical
Unit
Number

↑
Cartridge
Reference

↑
Specifies
Private,
Group, or
System
Cartridge



Accessing Devices

Data stored in files is initially input from other devices such as a terminal or magnetic tape. Each of these devices is assigned a LOGICAL UNIT NUMBER by the System Manager when the system is generated. Logical unit numbers are used to identify the device. Disc cartridges are also assigned logical unit numbers. However, the File Manager uses the cartridge reference in addition to or in place of the LU number.

The system also predefines session logical unit numbers that can be used to access any session device. Thus, the terminal you are using (no matter which terminal it may be) can always be referenced as session logical unit number 1.

CREATING FILES

Disc file creation always means that the file is given a name, is assigned an area on a disc cartridge and that a directory entry is made for the file.

One way to create a file and simultaneously store data into it is by using the STORE command.

```
:STORE,1,&FILE:CG:26
```

The STORE command creates a file that will contain ASCII coded information. The file in the example above is given the name &FILE; is assigned a security code of CG (that protects it from being written into by other users); allocates the disc space for the file on cartridge 26 and makes the directory entry. Data is input from the terminal (specified by the 1) following entry of the command:

```
:STORE,1,&FILE:CG:26  
THIS DATA IS TO BE STORED IN &FILE.
```

Each line of data is terminated by pressing the RETURN key (or ENTER key from a multipoint terminal). After the last line of data is input, the system must be told that there is no more forthcoming data. This is done by simultaneously pressing the CNTL and D keys (called a CONTROL-D) on a point-to-point terminal or by pressing the ENTER key once again on a multipoint terminal.

```
:STORE,1,&FILE:CG:26  
THIS DATA IS TO BE STORED IN &FILE.  
IT IS BEING INPUT FROM THE TERMINAL.  
Control-D or ENTER  
:
```

You can now list the contents of your file with the LIST command:

```
:LIST,&FILE
```

```
&FILE      T=00003 IS ON CR00026 USING 00001 BLKS R=0000
```

```
0001  THIS DATA IS TO BE STORED IN &FILE.
```

```
0002  IT IS BEING INPUT FROM THE TERMINAL.
```

```
:
```

Another way to create files is by using the Interactive Editor (EDITR). The Editor allows the user to create files that contain ASCII coded information. The Editor is invoked by entering the command:

Note that the commands (STORE, RUN, etc.) are spelled out for your convenience. You can enter only the first two characters for each command. For example, ST and RU can be used for STORE and RUN respectively.

```
:RUN,EDITR
```

The Editor will then respond with:

```
SOURCE FILE?
```

```
/
```

The slash (/) is the Editor's prompt character and is similar to the colon prompt. It is the Editor prompt which indicates that the Editor is waiting for input. By entering a zero (0) when this prompt is first received, the Editor will begin working with an empty file.

```
:RUN,EDITR
```

```
SOURCE FILE?
```

```
/0
```

```
EOF
```

```
/
```

You may now proceed to enter data preceding each line with a space:

```
/ THIS IS THE FIRST LINE.  
/ THIS IS THE SECOND LINE.  
/ THIS IS THE THIRD LINE.  
/ THIS IS THE LAST LINE.
```

After all the data has been input, you must give the Editor your selected file name and instruct the Editor to create it. For example, to store the above data in a file called &FILE2, enter:

```
/EC&FILE2
```

The EC tells the Editor that you have finished and that you want to End the process by Creating the file named &FILE2. The Editor will come back and tell you that it has followed your instructions and returned you to the system.

```
END OF EDIT  
:
```

You can now list the file shown previously using the LIST command:

```
:LI,&FILE2  
&FILE2  T=00004 IS ON CR00026 USING 00001 BLKS R=0000  
  
0001  THIS IS THE FIRST LINE.  
0002  THIS IS THE SECOND LINE.  
0003  THIS IS THE THIRD LINE.  
0004  THIS IS THE LAST LINE.
```

MODIFYING FILES

File modification is accomplished through the Interactive Editor. The Editor provides the capabilities for inserting, deleting and replacing lines of text. It will also insert, delete and replace characters within a line.

Invoke the Editor, but this time specify the previously created file to be used instead of a new file:

```
:RUN,EDITR
SOURCE FILE?
/&FILE2
  THIS IS THE FIRST LINE.
/
```

If a security code was specified for the file when it was created it should be specified after the file name in order to be able to update the same file (i.e., &FILE2:CG)

The Editor displays the first line in the file and then prompts for a command. The line displayed is called the PENDING LINE, and will remain so until the Editor is instructed to make another line the pending line.

Line Edits

You can use line edits to insert, delete, or replace a complete line of text. The commands used are a "R" to replace the pending line; an "I" to insert a line before the pending line; a space to insert a line after the pending line; and a minus (-) to delete the pending line.

To replace the pending line, use "R" as follows:

```
/P
  THIS IS THE FIRST LINE.
/RTHIS IS LINE NUMBER ONE.
/P
  THIS IS LINE NUMBER ONE.
/
```


The "P" command instructs the Editor to display the pending line. The "R" command allows you to replace the pending line with the new line entered.

To insert a line of text before the pending line, use the "I" command as follows:

```
/P
  THIS IS LINE NUMBER ONE.
/ITHIS IS THE BEGINNING OF THE FILE.
/
```

The line is inserted into the file before the pending line, but the pending line is not changed.

```
/P
  THIS IS LINE NUMBER ONE.
/
```

To show the INSERTED line, the pending line must be changed. This can be done by using the  (up arrow) command as follows:

```
/▲
  THIS IS THE BEGINNING OF THE FILE
/
```

This line now becomes the PENDING line. To make the line following the CURRENT pending line the NEW pending line, use the "/" (slash) command as follows:

```
//  
  THIS IS LINE NUMBER ONE.  
/
```

To insert a line after the pending line, use "space" as follows:

```
/ THIS LINE IS BEING INSERTED AFTER LINE ONE.  
/
```

This new line now becomes the pending line.

```
/P  
  THIS LINE IS BEING INSERTED AFTER LINE ONE.  
/
```

Before continuing, take a look at the entire file to see how it has changed. Since you are not at the beginning of the file, you must make the first line the pending line as follows:

```
/1  
  THIS IS THE BEGINNING OF THE FILE.  
/
```

This command is used to make the specified line (in this case the first line) the pending line. To list all or part of the file, use the "L" command as follows:

```
/L99
THIS IS THE BEGINNING OF THE FILE.
THIS IS LINE NUMBER ONE.
THIS LINE IS BEING INSERTED AFTER LINE ONE.
THIS IS THE SECOND LINE.
THIS IS THE THIRD LINE.
THIS IS THE LAST LINE.
EOF
/
```

The "L" command instructs the Editor to list the number of lines specified after the L (99). Since there were NOT 99 lines in the file, the entire file was listed and an EOF displayed to indicate the end of the file.

To delete one line (the pending line), use the "minus" (-) command as follows:

```
/3
THIS LINE IS BEING INSERTED AFTER LINE ONE.
/-
THIS IS THE SECOND LINE.
/
```

Line 3 was made the pending line and then deleted. The Editor displays the NEW pending line after the deletion.

If several successive lines were to be deleted, the number of lines could be specified after the minus (-) character to delete them with one command.

Character Edits

Characters in the pending line can be changed with character edits. Various control characters are used to insert characters into the pending line, delete characters from the pending line, replace characters in the pending line, and to truncate characters from the end of the pending line. Control characters CAN NOT be used from a Multipoint terminal. If you are using a Multipoint terminal, please skip the next section.

Point-To-Point Terminal Character Edits

Character edits are accomplished by using the "P" command in conjunction with the current delimiter, the slash (/), and control characters.

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

```
/2 THIS IS LINE NUMBER ONE.  
/P////////THE FIRST LINE,  
      ↑ Control-S entered here.  
  THIS IS THE FIRST LINE, LINE NUMBER ONE  
/
```

The current delimiter, /, is used to space over characters that are not to be changed. The Control-S, (press CNTL and S together) indicates where the following characters are to be

inserted. However, note that the Control-S is a non-printing character (as are all control characters) and will not be seen. Control-S will cause a return on some terminals.

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

```
/P
THIS IS THE FIRST LINE, LINE NUMBER ONE.
/P//////////-----
      |
      |Control-C entered here.
THIS IS THE FIRST LINE.
/
```

Again, the current delimiter is used to keep characters unchanged, Control-C indicates the beginning of the deletion, and the minus signs are used as place holders for the characters to be deleted. Pressing the RETURN key terminates the deletion before the period (.).

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

```
/Z
THIS IS THE SECOND LINE.
/P//////////LINE NUMBER TWO IN THIS FILE.
THIS IS LINE NUMBER TWO IN THIS FILE.
/
```

The current delimiter spaces over characters that are not to be changed and the replacement characters are typed where desired.

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

```
/P THIS IS LINE NUMBER TWO IN THIS FILE.  
/P /////////////////////////////////// ← Control-T and CR entered here.  
THIS IS LINE NUMBER TWO  
/
```

All of the character edit functions can be used in the same line:

```
/P THIS IS LINE NUMBER TWO  
/P //////////////////////////////////, IN THIS FILE.  
      ↑ Control-S entered.      ↑ Control-R entered.  
THIS IS THE SECOND LINE, LINE NUMBER TWO, IN THIS FILE.  
/
```

Control-R is used to stop the insert function and continue a replacement function.

Multipoint Terminal Character Edits

Character edits are accomplished by using the Q command in conjunction with the INSERT CHAR and DELETE CHAR keys on the terminal.

The Q command displays the pending line so it may be modified.

To insert characters into the pending line, issue the Q command and then use the INSERT CHAR key to enable character insertion as illustrated below:

```
/2 THIS IS LINE NUMBER ONE.
```

```
/Q THIS IS THE FIRST LINE, LINE NUMBER ONE.
```

↑ ↑

— INSERT CHAR key pressed — INSERT CHAR key pressed

```
THIS IS THE FIRST LINE, LINE NUMBER ONE.
```

```
/
```

Before pressing the ENTER key, move the cursor (the blinking underscore) beyond the end of the line. Otherwise, you will unintentionally truncate a line.

Deleting characters is accomplished in much the same fashion as modifying them. Issue the Q command, use DELETE CHAR to delete the unwanted characters, move the cursor beyond the end of the line and press the ENTER key as follows:

```

/P THIS IS THE FIRT LINE, LINE NUMBER ONE.
/Q THIS IS THE FIRST LINE, LINE NUMBER ONE.
    ↑────────────────────────────────────────↑
    DELETE CHAR key pressed
THIS IS THE FIRST LINE.
/

```

Character replacement with multipoint terminals is very simple. Simply retype the characters in the line as shown below:

```
/P
THIS IS THE SECOND LINE.
/Q
THIS IS LINE NUMBER TWO IN THIS FILE.
      retyped characters
THIS IS LINE NUMBER TWO IN THIS FILE.
/
```

As you may have already discovered, truncating lines is very easy. Simply move the cursor to the beginning of the character string you want to truncate and press the ENTER key as follows:

```
/P
THIS IS LINE NUMBER TWO IN THIS FILE.
/Q
THIS IS LINE NUMBER TWO IN THIS FILE.
      ^_____cursor positioned here
THIS IS LINE NUMBER TWO
/
```

All of these functions can be combined together in the same line to achieve the final results you want.

SAVING THE MODIFIED FILE

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

To save the modified file in a new file, terminate the Editor using the EC command. The new file name must be specified as follows:

```
/EC&NFILE  
END OF EDIT  
:
```

To replace the old file with the modified version use the ER command (end and replace) as illustrated below. The file name need not be specified if there was no security code. If a security code was placed on the file when it was created and was not specified as part of the response to the SOURCE FILE? prompt, it must be specified here along with the file name.

```
/ER  
END OF EDIT  
:
```

CHANGING A FILE NAME

You can change the name of a file you have created by using the RENAME (RN) command. For example, rename the file &FILE2 to &TFILE:

```
:RN,&FILE2,&TFILE
```

You can now list &TFILE to show that it's contents are that of the old file &FILE2.

```
:LIST,&TFILE
&TFILE  T=00004 IS ON CR00026 USING 00001 BLKS R=0000

0001  THIS IS THE BEGINNING OF THE FILE.
0002  THIS IS THE FIRST LINE.
0003  THIS IS THE SECOND LINE.
0004  THIS IS THE THIRD LINE.
0005  THIS IS THE LAST LINE.

:
:
```

If a security code was assigned to the file, it must be specified to rename the file. The command would then be:

```
:RN,&FILE2:CG,&TFILE
```

The security code will remain for the file. Only the name will change.

DELETING FILES

Several files have been created throughout this section of the manual. To remove them from the system so that the disc space may be put to another use, use the PURGE command.

The first file created was named &FILE and was given a security code of CG. To purge this file, enter the command:

```
:PURGE,&FILE:CG
```

The other files can be purged in the same fashion.

```
:PURGE,&TFILE  
:
```

COMMAND SUMMARY

Allocate Cartridge

```
:AC,cartridge reference
```

List mounted cartridges available to this session (Cartridge List)

```
:CL
```

Create a file and store data from a terminal

```
:STORE,1,name:security code:cartridge reference
```

List a file

```
:LIST,name
```


Create a file using the Editor

```
:RUN,EDITR
SOURCE FILE?
/0
EOF
/ data entered here
/ECname:security code
END OF EDIT
```

Modify files

```
:RUN,EDITR
SOURCE FILE?
/name:security code
first line listed
/
```

a. List the pending line

```
/P
```

b. Replace the pending line

```
/Renter new line
```

c. Insert line above pending line

```
/Ienter new line
```

d. Insert line below pending line

```
/ enter new line
```

e. Make previous line pending line

`/P`

f. Make next line pending line

`//`

g. Make specified line pending line

`/n`

h. List the specified number of lines

`/Ln`

i. Delete the pending line

`/-`

j. Delete the specified number of lines beginning with the pending line

`/-n`

k. Insert characters into the pending line

Point-to-Point: `/P` and use Control-S

Multipoint: `/Q` and use INSERT CHAR



- l. Delete characters from the pending line

Point-to-Point: **/P** and use Control-C

Multipoint: **/Q** and use DELETE CHAR

- m. Replace characters in the pending line

Point-to-Point: **/P** and type new characters

Multipoint: **/Q** and type new characters

- n. Truncate the pending line

Point-to-Point: Use Control-T

Multipoint: Position cursor and press ENTER

- o. Save modifications in a new file

/EName:security code

- p. Save modifications by replacing old file

/ER

- q. Go back X lines

/^X

- r. To get to the end of the file

/FEOF

s. To find out the pending line number

/n

Be careful in using line numbers because the line numbering will change if you have added or deleted lines.

t. Abort Editor and return to system

/A

Change a file name

:RN,name:security code,new name

Delete a file

:PURGE,name:security code:cartridge reference

EXAMPLE SUMMARY

:MC,25

:CL

| LU | LAST TRACK | CR | LOCK | P/G/S |
|----|------------|-------|------|-------|
| 18 | 0202 | 00025 | | P |
| 19 | 0202 | 00047 | | G |
| 02 | 0175 | 00002 | | S |

:STORE,1,&FILE:CG:25

THIS IS DATA TO BE STORED IN &FILE.

IT IS BEING INPUT FROM THE TERMINAL.

Control-D or ENTER

:LIST,&FILE

&FILE T=00004 IS ON CR00026 USING 00001 BLKS R=0000

0001 THIS IS DATA TO BE STORED IN &FILE.

0002 IT IS BEING INPUT FROM THE TERMINAL.

:RUN,EDITR

SOURCE FILE?

/0

EOF

/ THIS IS THE FIRST LINE.

/ THIS IS THE SECOND LINE.

/ THIS IS THE THIRD LINE.

/ THIS IS THE LAST LINE.

/EC&FILE2

END OF EDIT

:LIST,&FILE2

&FILE2 T=00004 IS ON CR00026 USING 00001 BLKS R=0000

0001 THIS IS THE FIRST LINE.

0002 THIS IS THE SECOND LINE.

0003 THIS IS THE THIRD LINE.
0004 THIS IS THE LAST LINE.

:RUN,EDITR
SOURCE FILE?

/AFILE2

THIS IS THE FIRST LINE.

/P

THIS IS THE FIRST LINE.

/RTHIS IS LINE NUMBER ONE.

/P

THIS IS LINE NUMBER ONE.

/ITHIS IS THE BEGINNING OF THE FILE.

/P

THIS IS LINE NUMBER ONE.

/A

THIS IS THE BEGINNING OF THE FILE.

//

THIS IS LINE NUMBER ONE.

/ THIS LINE IS BEING INSERTED AFTER LINE ONE.

/P

THIS LINE IS BEING INSERTED AFTER LINE ONE.

/1

THIS IS THE BEGINNING OF THE FILE.

/L99

THIS IS THE BEGINNING OF THE FILE.

THIS IS LINE NUMBER ONE.

THIS LINE IS BEING INSERTED AFTER LINE ONE.

THIS IS THE SECOND LINE.

THIS IS THE THIRD LINE.

THIS IS THE LAST LINE.

EOF

/3

THIS LINE IS BEING INSERTED AFTER LINE ONE.

/

```

THIS IS THE SECOND LINE.
/2 THIS IS LINE NUMBER ONE.

/P////////THE FIRST LINE,
      ↑ Control-S entered.
THIS IS THE FIRST LINE, LINE NUMBER ONE.
/P//////////-----
THIS IS THE FIRST LINE.      ↑ Control-C entered.
/1
THIS IS THE SECOND LINE.
/P////////LINE NUMBER TWO IN THIS FILE.
THIS IS LINE NUMBER TWO IN THIS FILE.
/P////////// Control-T entered.
THIS IS LINE NUMBER TWO
/P
THIS IS LINE NUMBER TWO
/P////////THE SECOND LINE, //////////////, IN THIS FILE.
      ↑ Control-S Control-R ↑
THIS IS THE SECOND LINE, LINE NUMBER TWO, IN THIS FILE.
/ER
END OF EDIT
:RN,&FILE2,&TFILE
:LIST,&TFILE
&TFILE T=00003 IS ON CR00026 USING 00001 BLKS R=0000

0001 THIS IS THE BEGINNING OF THE FILE.
0002 THIS IS THE FIRST LINE.
0003 THIS S THE SECOND LINE, LINE NUMBER TWO, IN THIS FILE.
0004 THIS IS THE THIRD LINE.
0005 THIS IS THE LAST LINE.

:PURGE,&FILE;CG
:PURGE,&TFILE
:

```

Developing a Program

The usual sequence of operations in developing a program is:

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

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

RTE FORTRAN IV is a programming language that accepts source statements written according to the rules described in the RTE FORTRAN IV Reference Manual.

SOURCE PROGRAM REMINDERS

FORTRAN Control Statement

The first statement of a FORTRAN program must be the control statement. This statement describes the output to be produced by the RTE FORTRAN IV compiler. This statement has the form:

FTN4,L

and starts in column 1.

FORTRAN Program Statement

The first source code statement of the FORTRAN program must be the Program statement with the form:

PROGRAM name

where name is a 5-character identifier by which the program will be known. This name will be used later for relocation and subsequent execution. This statement starts in column 7.

End Statement

The last statement in a FORTRAN program is a source statement that contains **END*** starting in column 7. This statement indicates the end of input to the compiler.

File Name Convention

A convention is used at HP when naming program files. A source file is indicated by a & as the first character and a relocatable file name begins with a % character. These conventions are used in all HP supplied software.

WRITING A SOURCE PROGRAM

A source program is written by using the RTE Interactive Editor as described in the Using Files section. First of all, invoke the Editor by typing

```
:RUN,EDITR
```

in response to a colon prompt (:). The Editor will issue a SOURCE FILE? query. Enter a zero (0) to begin with an empty work area. Begin typing in your program after receiving the Editor prompt (/), preceding each line with a space.

```
:RUN,EDITR
SOURCE FILE?
/0
EOF
/ FTN4,L
/ C RTE FORTRAN IV EXAMPLE PROGRAM
/
/   PROGRAM TEST
/   DO 10 I=1,3
/   B=I
/   A=SIN(B)+COS(B)
/   WRITE(1,15)A
```

```
/ 15   FORMAT(" VALUE OF A = ",F5.2)
/ 10   CONTINUE
/      END
/      END$
/
```

Program TEST will calculate and print out the sum of the sine and cosine of 1, then of 2, then of 3; a fixed-point format is to be used for printing, with a field width of five-characters, rounded to two fractional digits.

The program is now in the Editor work area and may be edited if necessary. When you are satisfied with it, save the source program in a disc file with the EC command and terminate the Editor.

```
/EC&TEST
END OF EDIT
:
```

COMPILING A SOURCE PROGRAM

Use the Compile program to compile your source and create the relocatable code.

```
:RUN,COMPL,&TEST,1,%TEST
```

This command specifies that the file named &TEST contains the source program, the compiler output (listing) is to be printed at the terminal, and to create the file named %TEST and store the relocatable code in it.

You will then see the following things happen, as illustrated in the example below.

1. A four-digit page number is printed out, followed by the compiler name, the time of day, and the date.
2. Each line of the source program is printed out, preceded by a line number in a four-digit format.
3. Next, the identification information about the compiler is printed out, followed by several statistics about the compilation. If there are any errors, the number of errors is stated.
4. The **END\$** statement is printed indicating the end of the compilation and the colon prompt appears.

```
:RUN,COMPL,&TEST,1,%TEST
```

```
PAGE 0001 FTN. 2:14 PM THU., 24 MAY., 1979
```

```
0001 FTN4,L
0002 C RTE FORTRAN IVB EXAMPLE PROGRAM
0003      PROGRAM TEST
0004      DO 10 I=1,3
0005      B=I
0006      A=SIN(B)+COS(B)
0007      WRITE(1,15)A
0008 15  FORMAT("VALUE OF A = ",F5.2)
0009 10  CONTINUE
0010      END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00062 COMMON = 00000

PAGE 0002 FTN. 2:14 PM THU., 24 MAY, 1979

0011 END\$
:

LOADING A PROGRAM

The RTE Relocating Loader is used to load programs for execution in the RTE-IV system. To load your program, invoke the Loader with

:RUN,LOADR

The Loader will prompt you for input with

/LOADR:

Enter the RELOCATE command followed by the name of the file containing the relocatable code created by the compiler.

/LOADR: RELOCATE, %TEST

The Loader will respond by printing the name of the program, the program bounds, and another prompt for input.

TEST 32002 32077
/LOADR:

Your response to this prompt should be END. This allows the Loader to finish loading your program by setting up the necessary links to the required system subroutines. The name and bounds of each of these routines will be listed, followed by statistical information about the relocation, and the Loader termination message which states that your program is ready for execution and that the loader has terminated. The colon prompt will then appear.

:RUN,LOADR

/LOADR: RELOCATE,%TEST

TEST 32002 32077

/LOADR: END

| | | | | | |
|-------|-------|-------|-------------|-------------|--------|
| CLRID | 32140 | 32146 | 750701 | 24998-16001 | |
| FMTID | 32147 | 33442 | 24998-16002 | REV.1901 | 781107 |
| FMT.E | 33443 | 33443 | 24998-16002 | REV.1901 | 781107 |
| PNAME | 33444 | 33511 | 771121 | 24998-16001 | |
| REID | 33512 | 33636 | 92067-16268 | REV.1903 | 790316 |
| ERR0 | 33637 | 33726 | 771122 | 24998-16001 | |
| .SNCS | 33727 | 34070 | 780424 | 24998-16001 | |
| ..FCM | 34071 | 34105 | 750701 | 24998-16001 | |
| ER0.E | 34106 | 34106 | 750701 | 24998-16001 | |
| .CMRS | 34107 | 34172 | 780424 | 24998-16001 | |
| FRMTR | 34173 | 37617 | 24998-16002 | REV.1901 | 781107 |
| .CFER | 37620 | 37675 | 750701 | 24998-16001 | |

4 PAGES RELOCATED 4 PAGES REQ'D NO PAGES EMA NO PAGES MSEG

/LOADR:TEST READY AT 2:16 PM THU., 24 MAY, 1979

/LOADR:\$END

:

EXECUTING A PROGRAM

After your program has been successfully loaded, you are ready to execute it by using the RUN command.

```
:RUN,TEST
```

The output from the program should then appear followed by the colon prompt.

```
:RUN,TEST  
VALUE OF A = 1.38  
VALUE OF A = .49  
VALUE OF A = -.85  
:
```

PROGRAM ERROR HANDLING

If you have any errors in your program, you can correct the errors by using the Editor. You must then compile, load, and execute the new version of your program. However, before loading the new version you must remove the old version of the program from the system. This is done by issuing the system OFF command.

```
:OFF,TEST  
:
```

3-6

If the old version is not removed from the system, the Loader will report a duplicate program when the new version is loaded.

COMPILE AND LOAD A PROGRAM

You may want to compile and load the program in one single step. This can be done with the CLOAD program. After you have created (or corrected) the source program using the Editor, run the CLOAD program. The following parameters are entered in order: source file name, list device, relocatable file name, and an optional control statement. The control statement provides you with the option of overriding the control statement in the source file if needed. See the following example of compiling and loading a sample program using the sample FORTRAN program we have created previously.

:RUN,CLOAD,&TEST

Note that the list device and relocatable file name are omitted. In this case, the list device is your session terminal and the relocatable file name is %TEST, following the Compiler Library conventions and default rules. If you want to specify the list device, enter the LU number of the required list device. You may also specify the relocatable file name (beginning with % as the first character). For example:

```
:RUN,CLOAD,&TEST,6,%XMP
```

Now Program TEST can be run.

:RUN,TEST

In case of errors, error messages will be displayed in the format shown below. Detailed description of the error messages is contained in the Terminal User's Reference Manual. If you use CLOAD after correcting a program, be sure to remove that program with the OFF command before running CLOAD.

```
/CLOAD:CL- XX
/CLOAD: <brie f error description>
/CLOAD:
```

where **XX** is a two-digit number

COMMAND SUMMARY

Creating a Source

```
:RUN,EDITR
SOURCE FILE?
/
EOF
/ FTN4,L
/ PROGRAM name
:
:
/
END$
/EName
END OF EDIT
:
```

Compiling the Source

```
:RUN,COMPL,source file,1,relocatable file
```

Loading A Program

```
:RUN,LOADR
/LOADR: RELOCATE,relocatable file
/LOADR: END
```

Compile and Load

```
:RUN,CLOAD, source file
```

BASIC PROGRAMS

Programs written in the BASIC language are executed under control of the BASIC Interpreter. They are not compiled or assembled like programs in the other languages. Complete information about BASIC commands and program statements is given in the Multi-User Real-Time BASIC Reference Manual. Therefore, this manual limits its discussion to accessing BASIC from RTE and the relationship of BASIC programs to File Manager files.

You can schedule BASIC with the RTE or FMGR RUN command. When you type RUN,BASIC, five parameters are set to default values as shown:

input
console
***RUN,BASIC,1,1,5,4,1** ← *error list*
 ↑
 list
 ↑
 output

BASIC accepts commands and program statements from the LU specified as the *console*, lists programs to the *list* LU (for BASIC, this is the console, not the line printer), and lists errors on the LU specified as the *error list* parameter. The *input* parameter defines the source of input to your BASIC program; *output* defines the device for output from it.

BASIC prompts for commands and program statements with the "greater than" symbol, >. If your program calls any external subroutines, you must enter the TABLES command first to declare the names of two tables used by BASIC, the Branch and Mnemonic Tables. The System Manager will be able to tell you the names of these tables which are created with a table generator program, RTETG. For a description of the content and purpose of these tables, see the Multi-User Real-Time BASIC Reference Manual. You do not need to know any more than the table names to create and run a BASIC program.

BASIC programs can be stored in File Manager files with the BASIC SAVE or CSAVE command and reloaded with the LOAD and RUN commands. The CSAVE command saves a semi-compiled form of the program. The LOAD command does not execute the program but merely makes it available for editing with the BASIC Interpreter. The RUN command runs the current program in memory if you do not provide a file name or loads and runs the program from a specified file. In BASIC, the file name and the program name are the same.

The following illustration shows one method for creating, saving, and running a BASIC program. You can also create programs with the File Manager or EDITR but the advantage of creating them with the BASIC Interpreter is that your statements are checked for correct syntax when they are entered. An example of creating a BASIC program with EDITR is given here. In the first example, the program contains a subroutine call to TIME, so the TABLES command must be used.

| | |
|----------------------------------------------------|-------------------------------------------------------------------------------------------|
| :RU,BASIC | ◀ Schedule BASIC |
| BASIC READY | ◀ BASIC indicates it is ready for a command or statement. Declare the BRT and MNT Tables. |
| >TABLES BRT,MNT | |
| >100 X = 987 * 123 | |
| >105 PRINT X | |
| >110 CALL TIME (X) | ◀ Type program statements. |
| >115 PRINT X; ``SECONDS AFTER MIDNIGHT`` | |
| >120 END | |
| >SAVE PROGX | ◀ Create file named PROGX and save program in it. |
| BASIC READY | ◀ Run current program. |
| >RUN | ◀ Print statement executed. X=121401. |
| 121401. | ◀ Subroutine call and print statement executed. |
| 50400.1 SECONDS AFTER MIDNIGHT | ◀ Type DEL to delete current program. |
| BASIC READY | ◀ Note:If you now type RUN, there are no statements to execute. |
| >DEL | ◀ If you type RUN with program name, PROGX is loaded and executed. |
| >RUN | |
| BASIC READY | |
| >RUN PROGX | ◀ BASIC waits for next command. |
| 12401. | ◀ Type BYE to exit and return to RTE. |
| 50430.2 SECONDS AFTER MIDNIGHT | |
| BASIC READY | |
| >BYE | |

Creating a Program Using BASIC Interpreter


```

:RU, EDITR
SOURCE FILES?
/0
EOF
/ 100 X = 987 * 123
/ 105 PRINT X
/ 110 END
/ECPRGZ
END OF EDIT
:RUN, BASIC
BASIC READY
>RUN PRGZ
121401.
BASIC READY
>

```

- ◀ Run EDITR.
- ◀ Space entered.
- ◀ EDITR indicates no lines in work area.
- ◀ Type space and new source statement.
- ◀ Create file named PROGZ.
- ◀ EDITR terminates.
- ◀ Run BASIC.
- ◀ Load and run PROGZ.
- ◀ PROGZ prints value of X.
- ◀ BASIC waits for next command or statement.

Creating a BASIC Program with EDITR

Executing A BASIC Program

```
>RUN,name
```

Removing A Program

```
>DEL,name
```

EXAMPLE SUMMARY

```
:RUN,EDITR  
SOURCE FILE?  
/0  
EOF  
/ FTN4,L  
/ C RTE FORTRAN IV EXAMPLE PROGRAM  
/ PROGRAM TEST  
/ DO 10 I=1,3  
/ B=I  
/ A=SIN(B)+COS(B)  
/ WRITE(1,15)A  
/ 15 FORMAT(" VALUE OF A = ",F5.2)  
/ 10 CONTINUE  
/ END  
/ END$  
/EC&TEST  
END OF EDIT  
:RUN,COMPL,&TEST,1,%TEST
```

PAGE 0001 FTN. 3:50 PM TUE., 2 JAN., 1979

```
0001 FTN4,L  
0002 C RTE FORTRAN IV EXAMPLE PROGRAM
```

```

0003      PROGRAM TEST
0004      DO 10 I=1,3
0005      B=I
0006      A=SIN(B)+COS(B)
0007      WRITE(1,15)A
0008 15    FORMAT(" VALUE OF A = ",F5.2)
0009 10    CONTINUE
0010      END

```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00062 COMMON = 00000

PAGE 0002 FTN. 2:14 PM THU., 24 MAY, 1979

0011 END\$

```

:RUN,LOADR
/LOADR:  RELOCATE,%TEST

```

```

TEST 32002 32077
/LOADR:  END

```

```

CLRID 32140 32146 750701 24998-16001
FMTID 32147 33442 24998-16002 REV.1901 781107
FMT.E 33443 33443 24998-16002 REV.1901 781107
PNAME 33444 33511 771121 24998-16001
REID 33512 33636 92067-16268 REV.1903 790316
ERR0 33637 33726 771122 24998-16001
.SNCS 33727 34070 780424 24998-16001
..FCM 34071 34105 750701 24998-16001
ER0.E 34106 34106 750701 24998-16001

```

.CMRS 34107 34172 780424 24998-16001
FRMTR 34173 37617 24998-16002 REV.1901 781107
.CFER 37620 37675 750701 24998-16001

4 PAGES RELOCATED 4 PAGES REQ'D NO PAGES EMA NO PAGES MSEG
/LOADR:TEST READY AT 2:16 PM THU., 24 MAY , 1979

/LOADR:END

:

:RUN,TEST

VALUE OF A = 1.38

VALUE OF A = .49

VALUE OF A = -.85

:OFF,TEST

:

Using Procedure Files

Groups of consecutive commands can be saved in a file known as a PROCEDURE file instead of entering each command at the terminal. This is particularly useful when the same set of commands is to be used over and over.

CREATING A PROCEDURE FILE

Procedure files are created, like source program files, with the Interactive Editor. To illustrate the function of procedure files, create a file to perform the program development steps discussed in Developing a Program Section.

```
:RUN,EDITR
SOURCE FILE?
/O
EOF
/ :RUN,COMPL,&TEST,1,%TEST
/ :RUN,LOADR,,%TEST,1
/ :RUN,TEST
```

Each command in a procedure file must be preceded by a colon (:). The last command should be a TRANSFER command.

```
/ :TRANSFER      or  / :TR      or  / ::
/ :ECPFILE
END OF EDIT
:
```

The file is then saved using the Editor **EC** (End and Create) command.

EXECUTING A PROCEDURE FILE

To execute a procedure file, use the TRANSFER command and specify the name of the procedure file.

:TRANSFER,PFILE or **:TR,PFILE** or **::PFILE**

As each command in the procedure file is executed the appropriate output is printed at your terminal and the command echoed.

:TRANSFER,PFILE
:RUN,COMPL,&TEST,1,%TEST

PAGE 0001 FTN. 4:16 PM TUE., 2 JAN., 1979

```
0001 FTN4,L
0002 C RTE FORTRAN IV EXAMPLE PROGRAM
0003     PROGRAM TEST
0004     DO 10 I=1,3
0005     B=I
0006     A=SIN(B)+COS(B)
0007     WRITE(1,15)A
0008 15  FORMAT(" VALUE OF A = ",F5.2)
0009 10  CONTINUE
0010     END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00061 COMMON = 00000
PAGE 0002 FTN. 4:16PM TUE., 2 JAN., 1979

0011 END\$

:RUN,LOADR,,XTEST,1
TEST 32042 32137

CLRID 32140 32146 750701 24998-16001
FMTID 32147 33442 24998-16002 REV.1901 781107
FMT.E 33443 33443 24998-16002 REV.1901 781107
PNAME 33444 33511 771121 24998-16001
REID 33512 33636 92067-16268 REV.1903 790316
ERR0 33637 33726 771122 24998-16001
.SNCS 33727 34070 780424 24998-16001
..FCM 34071 34105 750701 24998-16001
ER0.E 34106 34106 750701 24998-16001
.CMRS 34107 34172 780424 24998-16001
FRMTR 34173 37617 24998-16002 REV.1901 781107
.CFER 37620 37675 750701 24998-16001



4 PAGES RELOCATED 4 PAGES REQ'D NO PAGES EMA NO PAGES MSEG
/LOADR:TEST READY AT 2:32 PM THU., 24 MAY, 1979

/LOADR:\$END

:RUN,TEST
VALUE OF A = 1.38
VALUE OF A = .49
VALUE OF A = -.85
:TRANSFER
:

Note that the Relocating Loader prompted you for the name of the file to be relocated and **END** as in the Developing a Program Section. Another version of the command to invoke the Relocating Loader can be used to eliminate these responses. Remember to remove the old version of the program before you execute the new procedure file or an error will occur.

```
:RUN,EDITR
SOURCE FILE?
/PFILE
:RUN,COMPL,&TEST,1,%TEST
//
:RUN,LOADR
/P/////////,,%TEST,1
:RUN,LOADR,,%TEST,1
/ER
END OF EDIT
:OFF,TEST
:TRANSFER,PFILE
:RU,COMPL,&TEST,1,%TEST
```

PAGE 0001 FTN. 2:32 PM THU., 24 MAY , 1979

```
0001 FTN4,L
0002 C RTE FORTRAN IVB EXAMPLE PROGRAM
0003 PROGRAM TEST
0004 DO 10 I=1,3
0005 B=I
0006 A=SIN(B)+COS(B)
0007 WRITE(1,15)A
0008 15 FORMAT("VALUE OF A = ",F5.2)
0009 10 CONTINUE
0010 END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00062 COMMON = 0000

PAGE 0002 FTN. 2:32 PM THU., 24 MAY , 1979

0011 END\$

:RU,LOADR,,%TEST,1
TEST 32042 32137

| | | | | |
|-------|-------|-------|-------------|-----------------|
| CLRID | 32140 | 32146 | 750701 | 24998-16001 |
| FMTID | 32147 | 33442 | 24998-16002 | REV.1901 781107 |
| FMT.E | 33443 | 33443 | 24998-16002 | REV.1901 781107 |
| PNAME | 33444 | 33511 | 771121 | 24998-16001 |
| REID | 33512 | 33636 | 92067-16268 | REV.1903 790316 |
| ERR0 | 33637 | 33726 | 771122 | 24998-16001 |
| .SNCS | 33727 | 34070 | 780424 | 24998-16001 |
| ..FCM | 34071 | 34105 | 750701 | 24988-16001 |
| ER0.E | 34106 | 34106 | 750701 | 24998-16001 |
| .CMRS | 34107 | 34172 | 780424 | 24998-16001 |
| FRMTR | 34173 | 37617 | 24998-16002 | REV.1901 781107 |
| .CFER | 37620 | 37675 | 750701 | 24998-16001 |

4 PAGES RELOCATED 4 PAGES REQ'D NO PAGES EMA NO PAGES MSEG
/LOADR:TEST READY AT 2:32 PM THU., 24 MAY , 1979

/LOADR:\$END

:RU,TEST
VALUE OF A = 1.38
VALUE OF A = .49
VALUE OF A = -.85
:TRANSFER
:

COMMAND SUMMARY

Execute a Procedure File

:TRANSFER, name or :TR, name or ::name

EXAMPLE SUMMARY

```
:RUN,EDITR
SOURCE FILE?
/0
EOF
/ :RUN,COMPL,&TEST,1,%TEST
/ :RUN,LOADR
/ :RUN,TEST
/ :TRANSFER
/ECPPFILE
END OF EDIT
:TRANSFER,PFILE
:RUN,COMPL,&TEST,1,%TEST
```

PAGE 0001 FTN. 4:16 PM TUE., 2 JAN., 1979

```
0001 FTN4,L
0002 C RTE FORTRAN IV EXAMPLE PROGRAM
0003 PROGRAM TEST
0004 DO 10 I=1,3
0005 B=I
0006 A=SIN(B)+COS(B)
0007 WRITE(1,15)A
0008 15 FORMAT(" VALUE OF A = ",F5.2)
0009 10 CONTINUE
0010 END
```

FTN4 COMPILER: HP92060-1609 REV. 1805 (780310)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00061 COMMON = 00000

PAGE 0002 FTN. 4:16 PM TUE., 2 JAN., 1979

0011 END\$

:RUN,LOADR

/LOADR: REL,XTEST

TEST 32002 32076

/LOADR: END

| | | | | |
|-------|-------|-------|-------------|-----------------|
| CLRID | 32140 | 32146 | 750701 | 24998-16001 |
| FMTID | 32147 | 33442 | 24998-16002 | REV.1901 781107 |
| FMT.E | 33443 | 33443 | 24998-16002 | REV.1901 781107 |
| PNAME | 33444 | 33511 | 771121 | 24998-16001 |
| REID | 33512 | 33636 | 92067-16268 | REV.1903 790316 |
| ERR0 | 33637 | 33726 | 771122 | 24998-16001 |
| .SNCS | 33727 | 34070 | 780424 | 24998-16001 |
| ..FCM | 34071 | 34105 | 750701 | 24998-16001 |
| ER0.E | 34106 | 34106 | 750701 | 24998-16001 |
| .CMRS | 34107 | 34172 | 780424 | 24998-16001 |
| FRMTR | 34173 | 37617 | 24998-16002 | REV.1901 781107 |
| .CFER | 37620 | 37675 | 750701 | 24998-16001 |

4 PAGES RELOCATED 4 PAGES REQ'D NO PAGES EMA NO PAGES MSEG

/LOADR:TEST READY AT 2:16 PM THU., 24 MAY, 1979

/LOADR:\$END

:

```
:RUN,TEST
VALUE OF A = 1.38
VALUE OF A = .49
VALUE OF A = -.85
:TRANSFER
:RUN,EDITR
```

SOURCE FILE?

/PFILE

:RUN,COMPL,&TEST,1,%TEST

//

:RUN,LOADR

/P/////////,,%TEST,1

:RUN,LOADR,,%TEST,1

/ER

END OF EDIT

:OF,TEST

:TRANSFER,PFILE

:RUN,COMPL,&TEST,1,%TEST

PAGE 0001 FTN. 4:20 PM TUE., 2 JAN., 1979

```
0001 FTN4,L
0002 C RTE FORTRAN IV EXAMPLE PROGRAM
0003 PROGRAM TEST
0004 DO 10 I=1,3
0005 B=I
0006 A=SIN(B)+COS(B)
0007 WRITE(1,15)A
0008 15 FORMAT(" VALUE OF A = ",F5.2)
0009 10 CONTINUE
0010 END
```

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00061 COMMON = 00000

PAGE 0002 FTN. 4:20 PM TUE., 2 JAN., 1979

0011 END\$

:RU,LOADR,,%TEST,1
TEST 32042 32137

CLRID 32140 32146 750701 24998-16001
FMTID 32147 33442 24998-16002 REV.1901 781107
FMT.E 33443 33443 24998-16002 REV.1901 781107
PNAME 33444 33511 771121 24998-16001
REID 33512 33636 92067-16268 REV.1903 790316
ERRO 33637 33726 771122 24998-16001
.SNCS 33727 34070 780424 24998-16001
..FCM 34071 34105 750701 24998-16001
ERO.E 34106 34106 750701 24998-16001
.CMRS 34107 34172 780424 24998-16001
FRMTR 34173 37617 24998-16002 REV.1901 781107
.CFER 37620 37675 750701 24998-16001



4 PAGES RELOCATED 4 PAGES REQ'D NO PAGES EMA NO PAGES MSEG
/LOADR:TEST READY AT 2:32 PM THU., 24 MAY, 1979

/LOADR:\$END

:RU,TEST
VALUE OF A = 1.38
VALUE OF A = .49
VALUE OF A = -.85
:TRANSFER
:

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.

ERROR MESSAGE EXPLANATIONS

If a File Manager command cannot be interpreted by the system (input error) or has caused a recognized problem, an error message is printed in the form:

FMGR nnn or **FMGR-nnn**

where nnn is a three-digit number.

A brief description of the FMGR error can be requested with the ?? command.

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

The above case tells you that the requested file could not be found. You could also have requested an explanation by giving the error number following the ?? command.

```
:??,-6
FMGR-006 FILE NOT FOUND
```


If further assistance is needed in determining the cause of a particular error message, you can use the Help (HE) command. This command will provide a detailed explanation of the last error that occurred in your session or an explanation of a specific error.

```
:HE
FMGR-006
FILE NOT FOUND
ATTEMPT 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. This is known as Break Mode. The system indicates that you are in Break Mode by displaying a prompt in the form:

```
S = xx COMMAND ?
```

when any key is pressed on your terminal or when the ENTER key is pressed on a multipoint terminal. **xx** is the number assigned to your session when you logged on.

After receiving the Break Mode prompt, enter the ST command followed by the program name.

S = 17 COMMAND ? ST,TEST

90 0 0 0 0 0 0 0 0 /

↑
priority

↑
program state

↑
clock values



The system replies with information about the program. 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 DORMANT state as indicated by the zero (0). The other states are:

- 1 Scheduled - the program is scheduled and may be executing or waiting to execute
- 2 I/O Suspend - the program is suspended and waiting for an I/O operation to complete

- 3 General Wait - the program is in a "general wait" state suspended waiting for a system resource

- 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 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 Programs States summarizes the action you may take if your program is in a particular state.

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

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 all parameters are accurate and all devices to be used are operating and ready. |
| 2 I/O Suspend | Check the device to see that it is ready for use. You may have to wait for another program to finish using the device before your program can get access to 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 available memory. |
| 5 Disc Allocation Suspend | Check with the system manager as to why there is no system tracks available. You can try releasing tracks from your own programs or the EDITR program with the RT command. |
| 6 Operator or Program Suspend | Try typing the GO command to continue execution. |

5-4

OBTAINING SYSTEM STATUS

The current system status is obtained by executing the system program WHZAT. WHZAT describes the current system environment in two ways: by displaying all scheduled and suspended programs and their status or by displaying (in numeric sequence) all program partitions with their status.

To schedule the WHZAT program to display your programs and their status, enter the command:

```
:WH or S = xx COMMAND ? WH
```

The information returned is preceded by a heading that includes the current system time and column headings. Below is a sample output from the WHZAT program with explanations of the information provided.

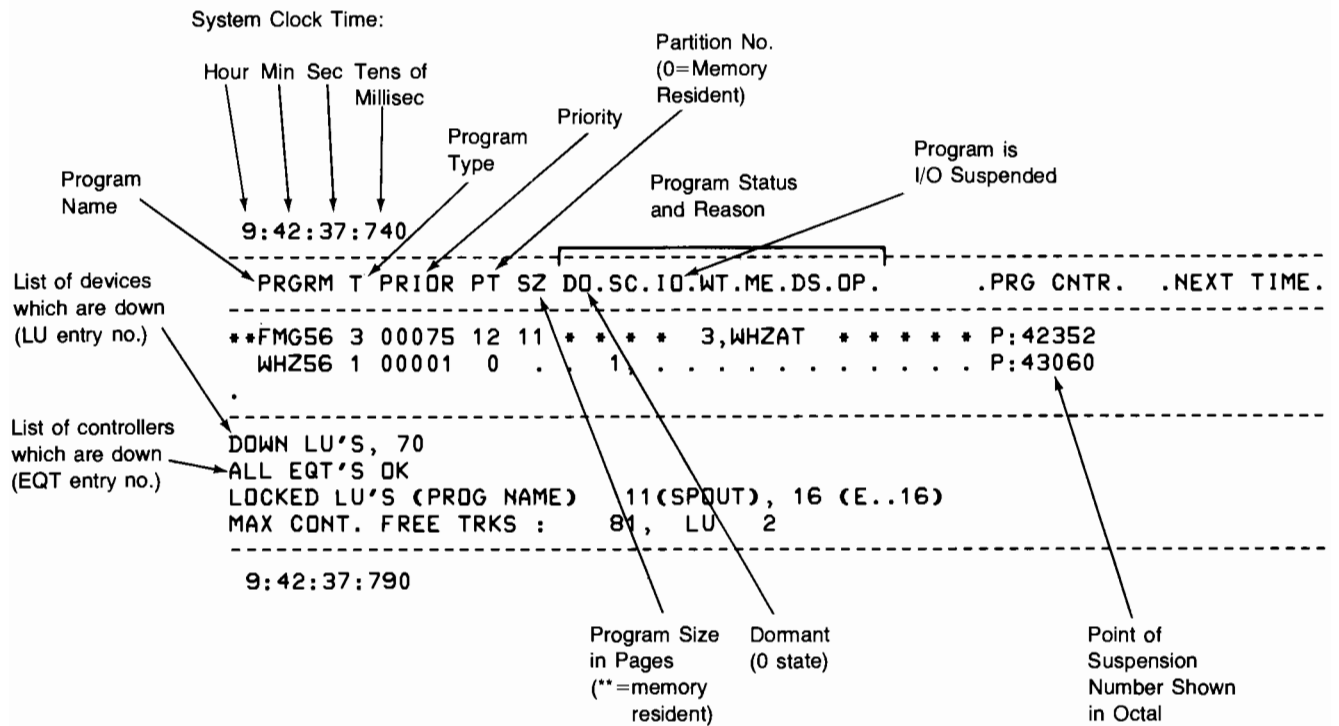
To schedule the WHZAT program to display all program partitions and their status, enter the command:

```
:WH,PA or S = xx COMMAND ? WH,PA
```

The information returned will appear in the format shown below.

Refer to the RTE-IVB Terminal User's Reference Manual for more information on the WHZAT program.

| | | | | | | |
|--------------------------------|-------------|------|------------------|-------------------------|-----------------------------------|--------------------------------|
| Hour | Min | Sec | Tens of Millisec | Partition Size in Pages | Background or Real-Time Partition | Program Currently in Partition |
| 9 | 43 | 23 | 540 | | | |
| Partition Number | PTN# | SIZE | PAGES | BG/RT | PRGRM | |
| 1 | 6 | 57- | 62 | RT | SMP | |
| 2 | 3 | 63- | 65 | BG | FLUSH | |
| 3S | 16 | 94- | 109 | BG | FMG58 | |
| 4S | 16 | 110- | 125 | BG | FMG01 | |
| 5 | 28 | 126- | 153 | BG | <NONE> | |
| 6M | 51 | 173- | 223 | BG | <NONE> | |
| . | . | . | . | . | . | |
| 11S | 19 | 173- | 191 | BG | <NONE> | |
| 12S | 16 | 192- | 207 | BG | DOWNR | |
| 13S | 16 | 208- | 223 | BG | FMG56 | |
| 14M | 128 | 224- | 351 | BG | <NONE> | |
| . | . | . | . | . | . | |
| 18S | 18 | 290- | 307 | BG | <NONE> | |
| 19S | 18 | 308- | 325 | BG | FMG59 | |
| 20S | 16 | 326- | 341 | BG | LOGON | |
| 21-30 | <UNDEFINED> | | | | | |
| 9:43:23:610 | | | | | | |
| Lowest and Highest Page Number | | | | | | |



Reason for I/O Suspend: EQT entry number (logical controller status $\left[\begin{array}{c} \text{octal content of} \\ \text{EQT status word} \end{array} \right]$)
logical controller status: 1 = device down, 2 = device busy, 3 = waiting for DMA channel

Reason for Wait:

| | |
|------------------------------------------|------------------------------------------------------------------------------------------|
| BL,EQT <i>eqt</i> | Buffer limit exceeded on the controller in EQT entry <i>eqt</i> |
| CL <i>ccc</i> | Waiting for class number <i>ccc</i> to complete GET |
| CLASS # | Waiting for a class number |
| LU/EQ DN | A device or controller is down. Look at DOWN LU'S or DOWN EQT'S list at bottom of report |
| LULK <i>lu</i> ,LKPRG = <i>prog name</i> | Logical unit with number <i>lu</i> is locked to named program |
| <i>program name</i> | Waiting for named program to complete |
| <i>program's</i> QUEUE | Waiting to schedule named program which is busy |
| RESOURCE | Waiting for resource number |
| RN <i>nnn</i> ,LKPRG = <i>prog name</i> | Resource number <i>nnn</i> is locked to named program |

Program Mode Abbreviations:

| | |
|-----------|----------------------------------------------------------------------------|
| 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 | Schedule (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 is 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 |



HEWLETT
PACKARD

Part No. 92068-90001
Printed in U.S.A. 4/80

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