

HP64000

**HP64000
Logic Development
System**



**System Software
Reference Manual**



CERTIFICATION

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Bureau of Standards, to the extent allowed by the Bureau's calibration facility, and to the calibration facilities of other International Standards Organization members.

WARRANTY

This Hewlett-Packard system product is warranted against defects in materials and workmanship for a period of 90 days from date of installation. During the warranty period, HP will, at its option, either repair or replace products which prove to be defective.

Warranty service of this product will be performed at Buyer's facility at no charge within HP service travel areas. Outside HP service travel areas, warranty service will be performed at Buyer's facility only upon HP's prior agreement and Buyer shall pay HP's round trip travel expenses. In all other cases, products must be returned to a service facility designated by HP.

For products returned to HP for warranty service, Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country.

HP warrants that its software and firmware designated by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

LIMITATION OF WARRANTY

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environment specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HP SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

ASSISTANCE

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.



OPERATING MANUAL



**SYSTEM SOFTWARE
REFERENCE MANUAL**

© COPYRIGHT HEWLETT-PACKARD COMPANY 1982,1983,1984
LOGIC SYSTEMS DIVISION
COLORADO SPRINGS, COLORADO, U.S.A.

ALL RIGHTS RESERVED

Manual Part No. 64980-90934

PRINTED: APRIL, 1984

Printing History

Each new edition of this manual incorporates all material updated since the previous edition. Manual change sheets are issued between editions, allowing you to correct or insert information in the current edition.

The part number on the back cover changes only when each new edition is published. Minor corrections or additions may be made as the manual is reprinted between editions. Vertical bars in a page margin indicates the location of reprint corrections.

First Printing February 1982 (P/N 64980-90913)
Reprinted July 1982
Second Edition January 1983 (P/N 64980-90928)
Third Edition April 1984 (P/N 64980-90934)

Table of Contents

Chapter 1. General Information

Introduction	1-1
Display Format	1-1
Directed Syntax Softkeys	1-2
Entering Commands	1-3
Re-executing Existing Commands	1-4
Recalling Commands	1-4
Editing Existing Commands	1-4
Stopping Execution	1-5
Command Files	1-5
Command Structure	1-6
Parameter Passing	1-6
Logging Commands	1-9

Chapter 2. File Manager Overview

Files	2-1
File Manager	2-2
File Security	2-3
Page Structure	2-4
Directory Format	2-4
User Libraries	2-4
File Management Commands	2-5

Chapter 3. System Commands

System Monitor	3-1
General Information	3-1
Variation of Monitor Softkeys with Hardware	3-1
Predefined Syntax Elements	3-4
<DATE>	3-5
<date qualifier>	3-6
<FILE>	3-7
<file group>	3-9
<file type>	3-10
<list destination>	3-11
<USERID>	3-12

Table of Contents (Cont'd)

System Command Syntax	3-13
append	3-14
assemble	3-16
<CMDFILE>	3-18
compile	3-20
copy	3-22
date&time	3-24
directory	3-26
edit	3-29
emulate	3-31
floppy	3-33
library	3-34
link	3-35
log_command	3-37
measurement_system	3-38
option_test	3-39
prom_programmer	3-40
purge	3-41
recover	3-42
rename	3-44
restore	3-45
run	3-48
state	3-50
store	3-52
terminal	3-55
timing	3-56
userid	3-58
verify	3-60
Chapter 4. Software Update Procedures	
Software Loading and Updating Procedure Using Flexible Disc Drive	4-1
Software Loading and Updating Procedure Using Tape Cartridge	4-2
Appendix A:	
Status and Error Messages	A-1
Subject Index	I-1

List of Illustrations

1-1. CRT Display	1-2
1-2. First Softkey Display	1-3

List of Tables

2-1. File Management Commands	2-5
3-1. Predefined Syntax Elements	3-4
A-1. General Status Messages	A-1
A-2. General Error Messages	A-5

NOTES

Chapter 1

GENERAL INFORMATION

INTRODUCTION

This manual describes commands available from the system monitor. The first two chapters cover general system information and attributes of the file system. Chapter 3 covers all system monitor commands and refers user to the separate editor, compiler, assembler/linker, emulation/analysis, state, timing and prom programmer manuals for details of their operation. Chapter 4 describes procedures for updating software. Appendix A may be used as a quick reference guide for general status/error messages associated with the system monitor.

DISPLAY FORMAT

The 64000 software package is oriented around the use of the development station CRT display as the sole system control device. The display is divided into four areas as shown in figure 1-1.

The STATUS line displays system status, explanatory messages, error messages and time. For the convenience of the operator, a quick reference status/error message summary for the monitor is given in Appendix A. For all other status/error messages refer to the appropriate module reference manual.

The command area consists of a line that wraps across three screen lines. Entered commands and interrogation responses are displayed in this line.

A blinking underline cursor is present in the top left-hand corner of the command line (just under the status line). The cursor indicates the current character position on the line.

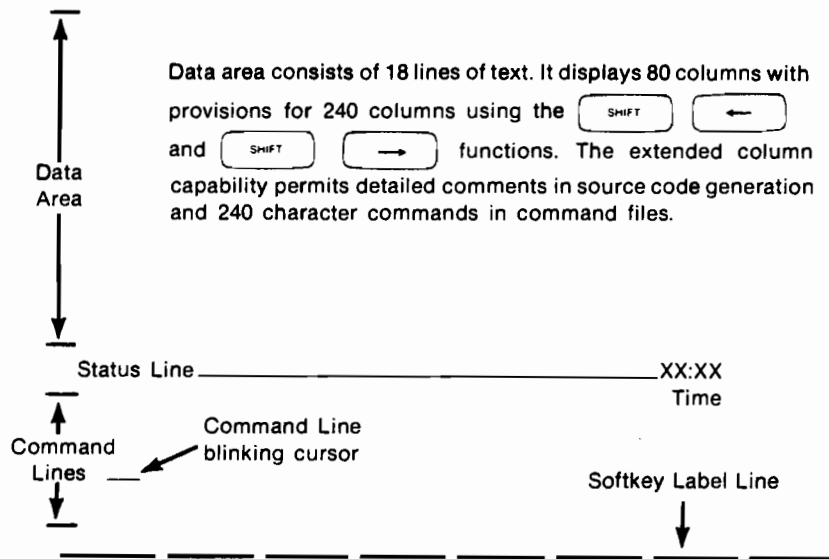


Figure 1-1. CRT Display

DIRECTED SYNTAX SOFTKEYS

Eight unmarked keys immediately below the CRT are labelled by the softkey label line just below the command line. These softkeys (see figure 1-2) indicate the complete set of allowable entries and change with each key stroke to reflect the next expected keyword or data in a command. If the user enters only that information prompted by the softkeys, the syntax will always be correct. Conversely, any entry not shown in the softkey labels will result in a syntactical error.

Note that the far right hand softkey is labelled **(---ETC---**). Pressing this softkey relabels the softkeys and presents additional softkey label line choices. Any item displayed on the CRT in all upper case and enclosed in angle brackets (<>) provides guidance for user-defined data. Press the softkey and information will be displayed on the status line as to requirements or defaults.

NOTE

It is also possible to type the commands (in lower-case characters) from the keyboard. This provides an alternative to the softkeys for the touch typist.

```
edit compile assemble link __*__ prom_prg run ---ETC---
```

where __*__ varies with hardware configuration.

Figure 1-2. First Softkey Display

To further enhance the use of the system, a uniform syntax/feature set was provided. For example, numeric constants may be specified in decimal, hexadecimal, octal, or binary throughout the system.

As each softkey is pressed, the cursor moves to the right providing reference for the inputs. If an error is made on the input, the cursor may be backspaced to the point of error. As the cursor moves back, the softkey label line information changes presenting the syntax choices available. If the cursor is backspaced to one of the key words, the key word may be replaced by pressing another softkey.

NOTE

A word may be replaced by positioning the cursor anywhere under that word and pressing another softkey. No other information in the command line is replaced or changed.

Entering Commands

System commands are composed of keywords (all lower case) and user-defined data. Softkeys are particularly valuable since they reflect all meaningful keywords available in building the command as well as prompts for user-defined data. The softkey labels change during the course of entering a command to guide the user through the command syntax.

Command syntax example:

```
store all_files :all_userids listfile printer
```

Once a command has been constructed in this manner, pressing the **RETURN** key causes execution to proceed. If a syntax error is encountered, a message is displayed and the cursor moves to the point of error.

The error may then be corrected, and pressing the **(RETURN)** key again will execute the command. When execution is successful, the cursor will appear left-justified in the command text.

Re-executing Existing Commands

After execution of a command, the same command may be re-executed by pressing the **(RETURN)** key. In fact, any time the cursor is in the command line, pressing the **(RETURN)** key causes execution of the command currently appearing on the line.

Recalling Commands

As commands are executed, the command is placed in a memory stack. The command(s) can be displayed in the command area by pressing the **(RECALL)** key. The exact number of commands that may be recalled is variable. Once a command is recalled to the command line, it may be executed by pressing the **(RETURN)** key; or, once a command is recalled, it may be edited by moving the cursor to the keyword to be edited. Select another keyword to make the edit. After the edit is made, press the **(RETURN)** key for execution.

Editing Existing Commands

Following successful execution, as described above, the cursor will be left-justified in the command text. At this point, a right cursor move signals the start of an edit of the command text. Softkey labels will reflect alternatives to keywords under which the cursor is placed. All keyboard edit functions may be used. Pressing the **(RETURN)** key again executes the edited line.

NOTE

If, instead of moving the cursor after execution, a typing key is pressed, the line will be cleared and a new command entry begun.

Example:

```
store all__files :all__userids listfile printer
```

Moving the cursor under `all__files` begins the edit. The softkey labels now show:

```
<FILE> (all__files) ( ) ( ) ( ) ( ) (listfile) ( )
```

These are all valid alternatives to `all_files` in the original command. Thus, typing filename `MYFILE` changes the command line to:

```
store MYFILE :all_userid listfile printer
```

and pressing the `(RETURN)` key causes it to execute.

NOTE

This command will not be executed if a tape cartridge or floppy unit is not installed.

Stopping Execution

The `(RESET)` key may be used at any time to stop execution of the current task (editor, assembler, compiler, etc.) and return control to the system monitor. Resetting the system monitor, in turn, reinitializes the system monitor. To avoid accidental reset, the `(RESET)` key must be pressed twice in succession to cause a reset. When `(RESET)` is pressed the first time, the system will enter a "pause" state.

No action may be taken except a stop or continue from the paused state. Execution continues only after pressing any key (except `(RESET)`). For example, a listing to the data area may be occurring and a line of interest rolls into view. A pause operation will leave the line on screen for observation, and the listing will continue when the pause is ended by pressing any of the standard keyboard keys.

COMMAND FILES

Command files are source files (created in edit mode or by using the log command) consisting of valid system commands to be executed in sequence. Command files provide a convenient method of passing parameters and a method of structuring a series of commands to be evoked with a minimum amount of keystrokes. The commands start in the system monitor and end with the last command in the file.

The commands are all executed conditional upon no syntactical or semantic errors. If syntactical errors are encountered during execution, the sequence will abort at that point. In addition to valid system commands, a parameter declaration line may optionally precede commands in a command file.

Command Structure

Examples 1 through 4 (following) are examples of command files. Example 1 is a command file in which system commands are used without passing parameters. Notice that the system commands are typed in lower-case letters and the file names are typed to conform to given file names. This command file assembles, links, and emulates (with a previous emulation configuration) a new version of a program.

NOTE

In these examples the command files are generated while in the editor.

Example 1:

`assemble TRY 1`

`link PGRMABS`

`emulate C8080 load PGRMABS`

where:

`assemble TRY 1` - Initiates an assembly of the source file TRY 1.

`link PGRMABS` - Initiates linking of linker command file PGRMABS.

`emulate C8080 load PGRMABS` - Initiates emulation using emulator command file C8080 and loads PGRMABS.

Parameter Passing

Example 2 is a command file using parameter passing. Parameters are defined as follows:

Passed Parameters

ASCII text "passed" to a command file. Any continuous set of ASCII characters (no spaces may be passed).

Formal Parameters

The name by which the command file refers to passed parameters. These parameters are upper-case symbols preceded by an ampersand (&).

Formal parameters may be used in any system command in place of any ASCII text not containing spaces. The ASCII text passed (in a passed parameter) will be directly substituted for the formal parameter at the time the command file is executed. This is referred to as CALL BY NAME.

Example 2:

Command File Name CMD2

PARMS &SOURCE

assemble &SOURCE

link PGRMABS

emulate C8080 load PGRMABS

where:

- | | |
|----------------------------|--|
| PARMS | - Parameter declaration line. It is the start of the formal parameter statement. |
| &SOURCE | - Ampersand designates a formal parameter. Parameter begins with an upper-case letter. |
| assemble
&SOURCE | - Designates that the passed parameter be the file name to be assembled at execution time. |
| link PGRMABS | - Runs PGRMABS linker command file. |
| emulate C8080 load PGRMABS | - Initiates emulation using configuration file C8080 and loads PGRMABS. |

When CMD2 is typed and the **RETURN** key is pressed, the system will prompt:

Define parameter &SOURCE:

The system prompts for the parameters to be passed. Type in the file to be assembled. The command file will then run through the emulate command.

Alternately, the passed parameter may be entered following the command file name. So CMD2 FILE4 executes the command file with FILE4 substituted for &SOURCE. The space between CMD2 and FILE4 is the delimiter.

Example 3 is an example of passing parameters in an ASCII string. This could be used to edit an existing file, changing, for example, the declaration in an assembler source file.

Example 3:

Command File Name CMD3

```
PARMS &FILE  
  
edit &FILE  
  
replace "8080" with "8085"  
  
end
```

where:

PARMS &FILE - Formal parameter declaration line. When CMD3 and **RETURN** are entered, the system prompts:

Define parameter &FILE:

The file to be edited may be entered as the passed parameter.

If it is desired to pass the replacement for "8080" as a parameter, the command file should be set up as follows:

Command File Name CMD4

```
PARMS &FILE &REP 1  
  
edit &FILE  
  
replace "8080" with "&REP 1"  
  
end
```

In this command file, the replacement for "8080" is defined from the keyboard. The system prompts:

Define parameter &FILE:

Type in the filename.

Define parameter &REP 1:

Type in the replacement to 8080 and press **RETURN**.

The command file will execute, replacing the 8080 with the designated information.

A command file may call a second command file. For example, the command file CMD4 just created may be modified to call command file CMD2.

Example 4:

Command File CMD5

```
PARMS &FILE &REP 1  
  
edit &FILE  
  
replace "8080" with "&REP 1"  
  
end  
  
CMD2 &FILE
```

where:

PARMS &FILE &REP 1 - Formal parameter declaration line. When CMD5 and RETURN are entered, the system prompts:

Define parameter &FILE (&REP 1)

The file to be edited may be entered as the passed parameter.

Note that command file CMD2 is automatically called by command file CMD5. All formal parameters in CMD2 must be entered on the same line at the end of CMD5 where CMD2 was evoked.

Any number of command files may be chained in this manner; however, they may not be nested. That is, when control is transferred from one command file to the next, there is no return to the departure point in the original command file.

Logging Commands

Rather than typing into command files the commands to be executed, provisions are made whereby the log command may be specified. When the log command is used, all valid commands are written to a source file. The source file created may be used to reissue the same set of commands or be edited to provide a more general purpose command file using parameter passing.

NOTES

Chapter 2

FILE MANAGER OVERVIEW

FILES

A file is defined as a group of data stored in an ordered, identifiable way on the system storage media. In the 64000, files are identified by four attributes: file name, userid, logical unit (LU) number, and file type (e.g., FILE:USERID: 0:source).

64000 file names are governed by certain conventions. They must start with an upper-case alphabetic character (A thru Z). They may consist of a mixture of alphanumeric characters (A thru Z, 1 thru 0 and an underscore (_)) after the first letter and are limited to nine characters. If the filename is longer than nine characters, it is truncated to the first nine.

The user ID portion of a file must start with an upper-case alphabetic character and is limited to six characters. After the first letter, the other five characters may be alphanumeric. If a user ID of more than six characters is entered, it is truncated to six.

There are a variety of file types (defined below), and each file is automatically assigned a type during the specific application in which it is used.

asmb_db	This file is created by the state analyzer and is a data base containing information from a link_sym file and all of its component asmb_sym files.
asmb_sym	This assembler symbol file, generated by the assembler or compiler, gives the local symbols for each relocatable file linked together for use by the emulator.
absolute	An absolute file generated by the linker containing absolute object code.
comp_db	This file is created by the linker when requested and is a data base containing information from all of the comp_sym files associated with relocatables in an absolute file.
comp_sym	This file is created by the compiler when requested and contains the entire compiler symbol table for use in creating comp_db.
data	No system programs make any assumptions about the contents of type data files. They may be generated by the user using simulated I/O or by specifying this type of file as the output file for a user program written in Host Pascal.

emul_com	An emulator command file generated in the emulator module. This file type includes all the information necessary to initiate the emulator.
link_com	A linker command file generated by the linker module. This file type includes all the information necessary to describe the relocation and linking process to the linker.
link_sym	This linker symbol file, generated by the linker, gives the global symbols used in a specific link operation for use by the emulator.
listing	A listing file generated to save the output of any command having a listfile option.
prom	A file generated by older versions of the prom programmer. This type of file is no longer produced (see Prom Programmer Manual).
reloc	A relocatable file generated by either the assembler or compiler containing relocatable object code.
source	A file generated by the editor.
system	This file appears under userid HP and identifies files containing system software. These file names are lower case and cannot be accessed or modified by the user.
trace	This file is created by some emulators and state/timing analyzers for saving results and specifications from analysis.

A given series of files may all have the same name but each file must be a different type. A given series of files may have the same name and type but reside on different bus discs.

Each file has a modified and accessed date placed on it by the file manager. This categorizes files for ease in backup. The date&time command should be used at power up so that these file dates are at the actual date and time.

FILE MANAGER

The file manager (a module of the operating system) is responsible for the organization and control of files. File management functions such as purge, recover, and rename are accomplished by the file manager.

As files are accessed (viewed) or modified on the 64000, the file manager automatically corrects the modified and accessed dates. For dates to be valid, the date&time command must be used correctly.

Unique to the 64000 is the ability to recover purged files. The recoverable files directory limits the maximum number of recoverable files to 127 on a hard disc and 63 on a floppy disc. When a file is purged, its space is linked to the available disc space. As more disc space is needed, the purged file will eventually be written over. Also, if the space reserved for a recoverable file is needed for an active file, the number of recoverable files will be reduced. The purge operation works on a first-in, first-written-over basis.

When a file is purged, its disc space is linked to the available disc space and deleted from the active file disc space. A recoverable directory entry will be made (this also allows a directory of all available recoverable files). The direction of disc space use is such that all available unwritten space is used before the purged files are written over. On the recover command, the purged file is linked to the used disc space and that file's entry is removed from the recoverable directory.

Each time a file is edited, the previous version is also put in the recoverable file list. It is therefore possible to recover all versions of a file that have not been written over. Since each recoverable file has a last access/modified date, the proper version may be quickly selected.

File Security

The user may protect his files by assigning a userid password to the files. Protected userids are not accessible from any other userid. Although protected userid files will be displayed with other files in a directory command, they will only be displayed with "all_userid" specifiers. The files can only be accessed after the default userid has been set to the protected userid. Once entered, the password on the protected userid cannot be later identified. It may be changed, however, while operating under the protected userid by reentering the userid command.

The 64000 operating system allows the designated system manager to list the passwords of all userids on the system by using a system file called `PASSWORDS:pMgR`. The protected userid `MgR` is intended to be the system manager's userid. Immediately after booting the operating system, the system manager should enter the monitor and assign a password by typing:

```
userid MgR password X
```

where X is the password of the manager's choice.

From userid "pMgR", the system manager may now enter the special command "prog `PASSWORDS`" and a listing of all protected userids on the system disc together with their passwords, will be printed to the printer, if any, or displayed, if no printer is found.

NOTE

Passwords may contain control characters, so the printer should be set to "display functions" or, the printer may be removed from the system and the control characters will be shown on the display.

The system manager should take care to remember the password on userid "pMgR" as that will always be necessary to get into the system manager's userid to issue the special command which lists all passwords.

Page Structure

The file management system has a linked list structure. Each of the files consists of a block of sectors referred to as pages. The number of sectors per page is constant for a given storage media but may vary to optimize certain file management operations. The pages of a file are linked in both forward and backward directions.

While a file is being updated the same sectors on the storage media are used. If the size of the file is increased, the file manager allocates another page to the file, linking it to the end of the last page.

Directory Format

The 64000 contains a directory which is organized as a hash coded list. Hash coding minimizes the amount of searching required to locate the directory entry for a given file. The hashed value of the file name indicates the directory sector on which the file information is most likely to reside. The data on that sector will indicate if the file exists or if another directory sector should be searched.

Each directory entry gives the name, user identification, and type of the file. Each entry contains pointers to the first and last pages of the file. In addition, two dates and times are kept for each file. One is the date and time that the file was last accessed. This is modified with the system date and time whenever the file is opened. The other is the date and time that the file was modified.

User Libraries

Libraries are a collection of relocatable modules that are stored on the system disc and can be referenced by the linker.

If a library file name is given as a response to the linker's "object files?" query, all the relocatable modules in the library file will be relocated and linked. If a library file name is given as a response to the "library files?" query, only those relocatable modules that define the unsatisfied externals will be relocated and linked. The remaining relocatable modules in the library file are ignored. It is possible to combine relocatables into a library by using the system library command.

FILE MANAGEMENT COMMANDS

All 64000 files are either disc or predefined device files. Disc files are specified by the user. Table 2-1 presents a listing of file management commands and their uses. The syntax for these commands is listed alphabetically under System Command Syntax in Chapter 3 of this manual.

Table 2-1. File Management Commands

File Mgmt. Command	Purpose
append	Used to append file to end of files on tape.
<CMDFILE>	Used to indicate a command file operation.
copy	Used to move a file from one destination to another.
date&time	Used to set or display date and time.
directory	Provides listing of files on bus disc, tape, or floppy disc.
library	Used to append one relocatable file to another.
log	Used to log commands to file.
purge	Purges file(s) from system.
recover	If a file is available, it allows that file to be recovered from a purged state.
rename	Changes file(s) from one name to another.
restore	Copies file(s) from tape or floppy disc to bus disc.
store	Copies file(s) from bus disc to tape or floppy disc.
userid	Used to establish user identification.
verify	Compares bus disc file(s) to tape or floppy disc file(s).

NOTES

Chapter 3

SYSTEM COMMANDS

SYSTEM MONITOR

General Information

This chapter is a reference guide to all commands issuable from the system monitor. Other subsystems (e.g. the editor) have their own monitors and the commands available in them are described in their subsystem reference manuals.

First will be a discussion of commands on system monitor softkeys as they vary with mainframes and optional hardware; then appear seven definitions for frequently used syntax elements; and finally, all system monitor commands are described in alphabetical order for quick reference.

Variation of Monitor Softkeys with Hardware

The first level of softkeys in the system monitor appear as:

(edit) (compile) (assemble) (link) (*) (prom_prog) (run) (===ETC===)

where (*) varies with hardware as follows:

- | | |
|------------|---|
| (meas_sys) | - appears when two or more hardware options for micro-processor system measurement are installed in the mainframe. These hardware options include emulators, state analyzers, timing, and software performance analyzers. |
| (emulate) | - appears when a single emulator control board is installed in the mainframe and no state, timing, or software performance analyzer. |
| (state) | - appears when a single state analyzer control board is installed in the mainframe and no emulator, timing, or software performance analyzer. |
| (timing) | - appears when a single timing analyzer control board is installed in the mainframe and no emulator, state, or software performance analyzer. |

The second level of softkeys (reached by pressing the (===ETC===) key from the first level) does not vary with hardware options and always appears as:

(directory) (purge) (rename) (copy) (library) (recover) (log) (===ETC===)

Since the directory command allows listing of the directory on the system disc as well as on local storage media, the softkeys reached following it vary with the local storage options. They appear as:

(<C|S|FILE>)(anychar)(anystring)(all_users)(rec_files)(*_*)(***)(listfile)

where (****) varies as follows:

- (****) - unlabelled in the case of no local storage or stand-alone operation from a floppy disc.
- (tapefiles) - appears when "DC100" mini-cartridge tape drive is installed in the mainframe.
- (floppy) - appears when dual mini-floppy drive is installed in the mainframe and station is on a hard disc system.
- (***) - unlabelled unless the system hard disc is a Model 7908, 7911, 7912, or 7914. In these cases, the softkey appears as (dc600) to allow the user to look at the directory of a "DC600" cartridge that is used as a streaming tape backup for these discs.

The copy command also varies with hardware configuration. When the mainframe is running standalone from a floppy disc, the rear panel HP-IB port is freed for HP-IB protocol use and the commands "copy FILE to hpib" and "copy hpib to ..." are permitted. The inner level softkey correspondingly vary showing the keyword "hpib" when the mainframe is running stand-alone from a floppy disc.

The third level of system monitor softkeys (reached by the (===ETC===) softkey from level two) appear as:

(userid)(date&time)(opt_test)(terminal)(<CMD|FILE>)(****)(*_*)(===ETC===)

where (****) varies, depending on whether a hard or floppy disc is the system disc, as follows:

[]

- nothing appears if the system disc is a hard disc and not a Model 7908, 7911, or 7912 and there is no local mass storage installed in the mainframe.

[BACKUP]

- appears if floppy or tape local mass storage is installed in the mainframe, or if the system hard disc is a Model 7908, 7911, or 7912.

[floppy]

- appears if the mainframe is running standalone from a floppy disc.

If the **[BACKUP]** softkey appears on the third level of softkeys, pressing it brings up a fourth level that appears in three different formats. If the mainframe has a tape option installed, the softkey display appears as follows:

[store] [restore] [append] [verify] [tension] [directory] [] [ETC]

If the mainframe has a floppy option installed, the softkey display appears as follows:

[store] [restore] [verify] [delete] [directory] [floppy] [] [ETC]

If the mainframe has no tape or floppy installed, the softkey display appears as follows:

[store] [restore] [] [] [] [directory] [] [ETC]

Note that the **[BACKUP]** softkey appears only on mainframes using Model 7908, 7911, or 7912 hard disc. These hard disc may provide a DC600 streaming tape backup; therefore, the store, restore, and directory functions only apply to the DC600 unit. In this circumstance, pressing these function keys will display: "store to__dc600", "restore from__dc600", and "directory dc600" respectively.

PREDEFINED SYNTAX ELEMENTS

The following syntax elements (table 3-1) are used frequently throughout the system commands that are listed in this chapter. They are predefined here to avoid extreme repetition.

NOTE

Angle brackets denote frequently used syntax elements which are predefined prior to command discussions.

Lower-case syntax elements are composed of keywords and user data. Upper-case syntax elements are composed only of user-defined data. For example, <list destination> is a predefined element which is used in all commands having a listing destination as part of their output.

Table 3-1. Predefined Syntax Elements

<DATE>	Used for calendar dates.
<date qualifier>	Used to reference files by date.
<FILE>	Used to reference bus disc files.
<file group>	Used to reference groups of files.
<file type>	Used to reference files by type.
<list destination>	Used to specify listing output.
<USERID>	Used to specify user identification.

The syntax descriptions of the predefined elements listed in table 3-1 are given on the following pages.

SYNTAX

<p>DAY/MONTH/YEAR</p> <p>or</p> <p>DAY . MONTH . YEAR</p>

Example:

2/1/83 Equivalent to calendar date 2 Jan 1983.



FUNCTION

Used to enter calendar dates into the system.

Parameters:

DAY	dd = day (1 through 31).
MONTH	mm = month (1 through 12). yy = year (two or four digits).
YEAR	

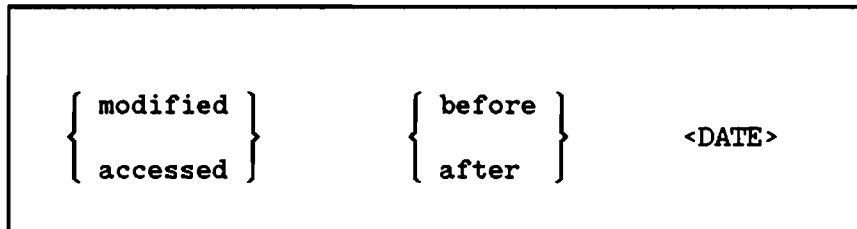
Calendar accurate from January 1, 1977 to year 2167.

DESCRIPTION

Dates are used in the system either in setting the date (see `date&time` command) or for referring to files by the date which they have associated with them (see `<date qualifier>`).

<date qualifier>

SYNTAX



NOTE

The date qualifier may be used if the time and date are not set. For long term projects the date and time should be set to provide accurate management of all files.

Example:

modified before 2/6/81 Refers to files modified before 2 June 1981.

FUNCTION

Selects files modified (written to) before or after a given date. Selects files accessed (read from) before or after a given date.

Parameters:

- | | |
|----------|---|
| modified | - Written to. |
| accessed | - Read from. |
| before | - Longer ago than (exclusive of date specified). |
| after | - More recent than (exclusive of date specified). |
| <DATE> | - see <DATE> syntax. |

DESCRIPTION

The <date qualifier> allows accurate and up-to-date file management control. As a file is accessed or modified, the current date is copied to the directory giving the latest information on that particular file.

SYNTAX

```
FILENAME [:USERID] [:DISC LU#]
```

Examples:

PROG1:MAX:1	File PROG1 with userid MAX on disc assigned logical unit (LU) # 1.
PROG1:	File PROG1 with the blank userid on all bus disc.
PROG1	File PROG1 with the current userid on all bus discs.
PROG1:0	File PROG1 with the current userid on bus disc LU 0.

REFERENCES

All commands involving disc files.

FUNCTION

Format used to identify the attributes of a file.

Parameters:

FILENAME	Up to 9 alphanumeric characters starting with an upper-case alphabetic character. If a file name greater than 9 characters is used, only the first 9 characters are significant. Underscore is the only special alphanumeric character permitted.
USERID	Up to 6 alphanumeric characters starting with an upper case alphabetic character. If a userid greater than 6 characters is used, only the first 6 characters are significant. Userid may be blank.

DISC LU #

When no logical unit number is specified, the default is to act on the file meeting name, userid, and type qualifiers on the bus disc of lowest LU value. For example, if source file A exists on bus discs 0 and 1, the "edit A" command will bring in A on bus disc 0. If A on bus disc 0 is later purged, the same command will bring in A on bus disc 1.

NOTE

Single bus disc users need not make reference to disc LU numbers.

DESCRIPTION

A file name is required for all files. File names are user definable and must be entered from the keyboard. Implied in all file names are :userid and :disc #.

The userid permits user code identification and a method of code management.

In multidisc systems, :DISC LU # is used to manage the placement and retrieval of code from different bus discs.

The following conventions are used when file names are displayed:

- Files are always displayed with file name and userid.
- The bus disc logical unit number is displayed only if it is nonzero.
- The type of the file is displayed if not implied by the context of the command.

SYNTAX

$\left\{ \begin{array}{l} \text{all_files} \\ \text{FILENAME} \end{array} \right\}$	$\left[\begin{array}{l} \text{:all_userids} \\ \text{:USERID} \end{array} \right]$	$[\text{:DISC \#}]$
--	--	---------------------

Examples:

all_files	References files of any name with the current userid on all bus discs.
all_files:all_userids	References files of any name with any userid on all bus discs.
F:MAX:1	References files named F with userid MAX on bus disc LU 1.
F:all_userids:0	References files named F with any userid on bus disc LU 0.

Parameters:

all_files	Files of any name.
all_userids	Files with any userid.
FILENAME	See <FILE> syntax.
USERID	See <USERID> syntax.
DISC LU #	See <FILE> syntax.

DESCRIPTION

The <file group> construct allows the use of two "wild card" specifiers in selecting groups of files to be used in a command. The *all_files* allows files of any name to be referenced. The *all_userids* allows files with any userid to be specified.

<file type>

SYNTAX

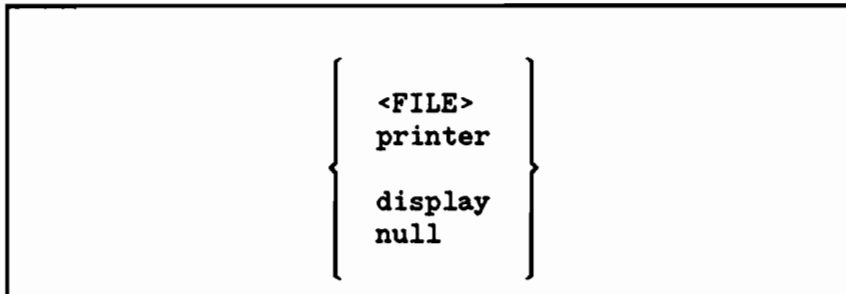
```
source  
reloc  
absolute  
listing  
emul_com  
link_com  
prom  
asmb_sym  
link_sym  
data  
trace  
comp_sym  
asmb_db  
comp_db
```

Examples:

BOB1:source	Source file named BOB1.
Math:reloc	File created by assembly of source file Math.
DM:absolute	File created by link of link__com file DM.

<list destination>

SYNTAX



Default Values

The default for <list destination> is specified by the user at the time the userid is entered. Thereafter, if no new destination is entered, the selected default is used.

Examples:

LIST	Specifies listing output to file LIST of type listing.
printer	Specifies listing output to line printer.

Parameters:

<FILE>	See <FILE> syntax. The type associated with this file is listing.
printer	Specifies output listing be printed at the line printer.
display	Specifies output listing be rolled on the display in the data area.
null	Specifies no listing output (except errors).

DESCRIPTION

The <list destination> allows selection of the file or device for output listing. In certain instances, the contents of such a listing output may be qualified by options contained within the command. Listing files may be viewed with the editor; but, since they are not source files, modification of listing files has no value except possibly for user convenience in viewing them.

<USERID>

SYNTAX



Default Value

USERID - current userid

Examples:

PROG1:MAX Refers to file PROG1 with userid MAX, all bus discs.

PROG1: Refers to file PROG1 with blank userid.

FUNCTION

Used to set userid.

Parameters:

USERID Up to 6 alphanumeric characters starting with an upper case alphabetic character. If a userid greater than 6 characters is used, only the first 6 characters are significant. Userid may be blank.

DESCRIPTION

Userids are provided primarily as a convenience in managing files. Once specified in a userid command, all file references made thereafter are assumed to be to files under the current userid. Thus, references to files of another userid must be explicit. This allows a free hand in naming files, as the same name may be used unambiguously under multiple userids.

SYSTEM COMMAND SYNTAX

The system commands are presented in alphabetical order on the following pages.

append

SYNTAX

```
*append [<file group> [:<file type>] [<date qualifier>]]  
        [listfile <list destination>]
```

*Only appears on systems with mini-cartridge
tape backup.

Default Values

<file group>	Defaults to files of any name with current userid on all bus disc.
<file type>	If no type is specified the types source, emul__com, link__com, and trace are assumed.
<date qualifier>	If no date qualifier is specified, dates are ignored and do not affect files being appended to tape.
<list destination>	Defaults to user specified listfile default. See userid command syntax.

Examples:

append	Appends to tape all files with current userid of type source, emul__com, and link__com.
append X:all userids	Appends to tape, all files named X:source with any userid of type source on all bus discs.
append all__files modified after 15/2/82	Appends to tape all files under current userid that have been "written to" since February 15, 1982.

FUNCTION

Append allows bus disc files to be added to an already existing tape without changing the files already on tape.

Parameters:

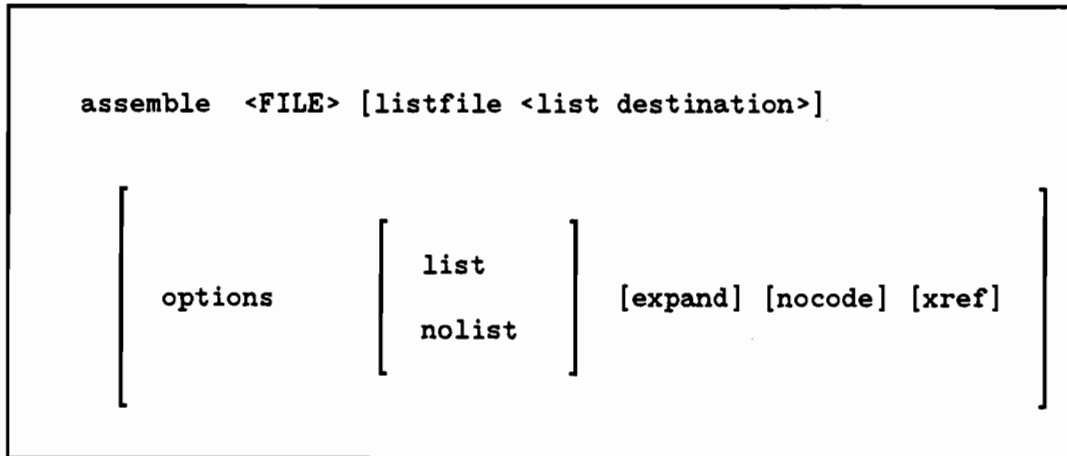
<file group>	File or files to be appended.
<file type>	Type of file or files to be appended.
<date qualifier>	Selects files by modified/accessed date.
<list destination>	Selects file or device for listing output.

DESCRIPTION

This command copies files to the end of a tape with existing files. The tape directory is updated with the correct file name after all designated files have been appended to the tape.

assemble

SYNTAX



Default Values

<list destination>

By default, listing output is sent to the listfile default specified in last userid command.

options

If options are not entered, all listings occur as per pseudo instructions specified in the source file. If options are specified, followed by nothing, then:

- Output listing of source program with object codes and error messages.
- No expansion of macros and multiple-byte pseudo instructions.
- No listing of the symbol cross-reference list.

Examples:

assemble SAM Assembles source file SAM; output listing to specified listfile default.

assemble SAM listfile CHARLEY

 Assembles source file SAM; output listing to file CHARLEY of type listing.

assemble SAM listfile display options nolist nocode

 Assembles source file SAM; producing no relocatable file and listing only errors to the display.

FUNCTION

The assembler translates source program inputs into relocatable object modules that can be linked and loaded into the emulator or programmed into PROMs. Absolute addresses are assigned by the linker.

Parameters:

- | | |
|---------------------------------|---|
| <FILE> | Source file to be assembled. Selects file or device for listing output. |
| <list destination> | See <list destination> syntax. |
| options | Allows user to override listing options specified in the source file. |
| list | Overrides all list and nolist pseudos in the source file and forces listing of all lines. |
| nolist | Overrides all list and nolist pseudos in the source file and forces no listing except errors. |
| expand | Overrides all list and expand pseudos in the source file and forces expanded list of all areas selected for listing in source file. |
| nocode | Causes assembly with no object code generation or relocatable file creation. |
| xref | Causes a cross-reference to be printed to the <list destination>. |

DESCRIPTION

Refer to the Assembler/Linker Reference Manual for a detailed description of the assembler.

<CMDFILE>

SYNTAX

<FILE> [PARMS]

Examples:

OD	File name OD, current userid
OD:Tom	File name OD, userid Tom
OD:Tom Prog Dash	File name OD, userid Tom with passed parameters Prog and Dash

FUNCTION

Typing a command file name in the command area and pressing **RETURN** begins execution of commands in file in sequence.

Parameters:

<FILE>	A command file is a source file that consists of a series of valid system commands.
PARMS	ASCII text "passed" to a command file. Any continuous set of ASCII characters (no spaces may be passed).

DESCRIPTION

Command files are source files (created in edit mode or by using the log command) consisting of valid system commands to be executed in sequence. Command files provide a convenient method of passing parameters and a method of structuring a series of commands to be evoked with a minimum amount of keystrokes. The commands start in the system monitor and end with the last command in the file.

The commands are all executed conditional upon syntactical or semantic errors. If syntactical errors are encountered during execution, the sequence will abort at that point. In addition to valid system commands, a parameter declaration line may optionally precede commands in a command file.

The parameter declaration line is used to pass variable parameters from a command file. The system will prompt for the actual parameter or the parameter can be typed in the command line along with the command file name.

Formal parameters may be used in any system command in place of any ASCII text not containing spaces. The ASCII text passed (in a passed parameter) will be directly substituted for the formal parameter at the time the command file is executed. This is referred to as CALL BY NAME.

Refer to Chapter 1 for several detailed examples of Command Files.

compile

SYNTAX

```
compile <FILE> [listfile <list destination>]
```

```

[
  options
  [
    list
    nolist
  ]
  [expand] [nocode] [xref] [comp_sym]
]

```

Default Values

<list destination>

By default, listing output is sent to the listfile default specified in last userid command.

options

If options are not entered, all listings occur as per pseudo instructions specified in the source file. If options are specified, followed by nothing, then:

- Output listing of source program with object codes, and error messages.
- No expansion to show generated assembly language.
- No listing of the symbol cross-reference list.
- No comp__sym file generation.

Examples:

compile PASCALPRG Compiles source file PASCALPRG with output listing to specified listfile default.

compile PASCALPRG listfile MYLIST Compiles source file PASCALPRG; listing to file MYLIST of type listing.

compile PASCALPRG listfile display options nolist nocode Compiles source file PASCALPRG producing no relocatable file and listing only errors to the display.

FUNCTION

Initiate compile function for source programs.

Parameters:

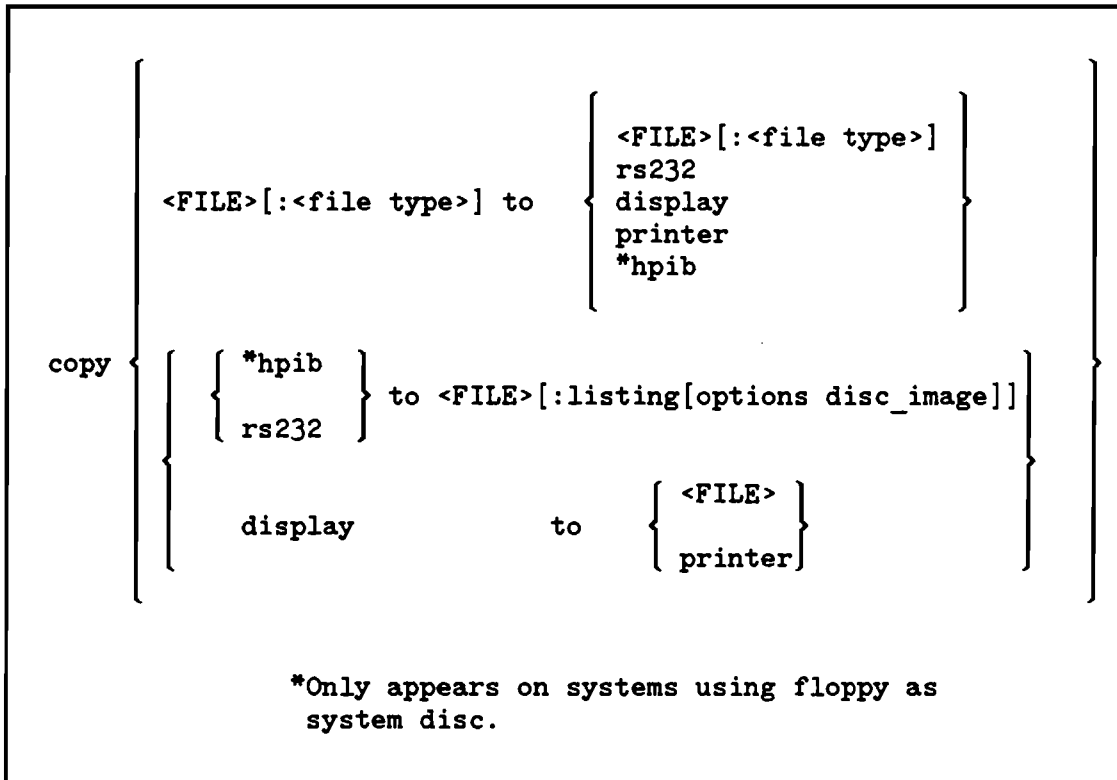
- | | |
|--------------------|---|
| <FILE> | Source file to be assembled. See <FILE> syntax. |
| <list destination> | File or device for listing output. |
| options | Allows user to override listing options specified in the source file |
| list | Overrides all list and nolist pseudos in the source file and forces listing of all lines. |
| nolist | Overrides all list and nolist pseudos in the source file and forces no listing except errors. |
| expand | Overrides all list and expand pseudos in the source file and forces expanded list of all areas selected for listing in source file. |
| nocode | Causes compilation with no object code generation nor relocatable file creation. |
| xref | Causes a cross-reference listing to be printed to the <list destination>. |
| comp__sym | This file is created by the compiler when requested and contains the entire compiler symbol table for use in creating comp__db (refer to link syntax for a description of comp__db file). |

DESCRIPTION

Refer to Compiler or Host Pascal Reference Manual for a detailed description of the compiler.

copy

SYNTAX



Default Values

<file type>	Defaults to type source or listing with source taking precedence.
:listing	If this is not specified, default is to same type file as that of the "from" file.
options disc_image	For files of type reloc, absolute, asmb_sym, and link_sym: the default format of a listing of these files decodes the data. This decoding can be overridden by options disc_image.

Examples:

copy PROG1 to printer	Prints source file or listing PROG1 to the line printer.
copy PROG1 to rs232	Writes source or listing file PROG1 out over RS-232 link.
copy rs232 to NEWFILE	Reads from RS-232 input and writes input into source file NEWFILE.
copy PROG1:absolute to PROG2	Copies file PROG1 type absolute into file PROG2 type absolute.

FUNCTION

Gives the user the capability of duplicating any file on bus disc by copying it to a file of another name. Also allows for communication with the RS-232 link for both input and output.

Additionally, the copy command is valuable in viewing files not normally observable by the user. By copying a non-ASCII file to a device or file of type listing, the user may obtain a readable version of the binary file's contents. In the case of relocatable, absolute, asmb__sym, and link__sym files, copy decodes somewhat the binary information in listing it (this decoding may be overridden using options disc__image). Otherwise, the binary data is output in hexadecimal and ASCII.

Parameters

<FILE>	Name of file to be copied from or to.
<file type>	Type of file to be copied.
rs232	Designates input or output over RS-232 link.
display	Designates input or output from the data area of the display.
printer	Designates output to the printer.
options disc__image	Overrides decoding of binary data forcing ASCII and hexadecimal dumps of all data in file.

DESCRIPTION

The copy command is used to move a file from one destination to another. When moving files through the RS-232 link, refer to the Installation and Configuration Manual for information regarding control signals, timing diagrams, and format requirements.

date&time

SYNTAX

```
date&time  [<DATE>] [HOURS:MINUTES]
```

Default Values

<DATE>	If no date is specified, the current system date is displayed on the status line.
HOURS:MINUTES	If no time is specified, the current system time is displayed on the status line.

Example:

<code>date&time 27/2/81</code>	Sets current system date to Wed, 27 Feb, 1981.
<code>date&time 13:15</code>	Sets time to 13:15 (24-hour clock).
<code>date&time 27.2.81 9:45</code>	Sets both date and time.

FUNCTION

Allows user to set the current date and time into the system.

Parameters:

<DATE>	Day of year.
HOURS	0-24.
MINUTES	0-59.

DESCRIPTION

The 64000 operating system maintains a clock/calendar for use by the various software development modules. All generated listings contain the current time and date as a heading. The time and date of creation or modification of files is also maintained and can be displayed in a directory listing of files. The clock is derived from the power line and automatically accounts for either 50-Hz or 60-Hz line frequency. The internal calendar is accurate from January 1, 1977 to the year 2167.

At power up, one development station can be used to set the time and date on all development stations; clock time is continuously displayed at the far right of the status line. If the time is not set, it starts at 00:00. It can be changed by issuing the date&time command. If the clock is reset on any development station, all stations will reflect that change. If the date is not set at power up, the system date defaults to the software revision date. If the development station is left on at all times and the date and time have been set, the correct date and time will be automatically maintained.

directory

SYNTAX

```

directory [
    < [C] [S] FILE GROUP>[:<file type>][<date qualifier>]
    all_userid[:DISC LU#][:<file type>][<date
        qualifier>]
    recoverable_files
    *tape_files
    **floppy_files
    ***dc600 [<file group>[:<file type>]]
]

[listfile <list destination>]

*Appears only on systems with tape option

**Appears only on systems with floppy option

***Appears only on systems with bus discs having
an integral tape backup. NOTE: discs requiring
multiple DC600 cartridges for backup do not permit
this function. No dc600 directory can be listed.

```

Default Values

<[C][S] FILE GROUP>	Default is all files under current userid.
<file type>	Default is all types.
<date qualifier>	Default is to ignore modified and accessed dates.
<list destination>	Defaults to user specified listfile default. See userid command syntax.



Examples:

directory List all files of all types on bus disc LU 0 with the current userid to the default listfile device or file.

directory A **[S]** Lists all files beginning with A under current userid.

directory all__files:all__userids:source

List all files of type source on bus disc LU 0 with any userid to the default list device or file.

directory all__files:SAM modified before 26/1/81

Lists all files of all types with userid SAM which were modified before 26 Jan 1981.

FUNCTION

Provides a selective listing of bus disc files, all files on a tape or floppy, or all available recoverable files.

Parameters

<**[C]****[S]** FILE GROUP> Files to be included in directory. Syntax is identical to that of <file group> except that the special symbols **[C]** (equivalent to any non-blank character) and **[S]** (equivalent to any string (includes blanks)) are allowed in FILENAME and USERID.

<file group> Files to be included in directory of dc600.

<file type> Type of files to be included in directory.

<date qualifier> Allows viewing of files by last modification or access date.

floppy__files Specifies a directory of all files stored on an inserted floppy.

tape__files Specifies a directory of all files stored on an inserted tape.

recoverable__files Specifies a directory of all recoverable files.

dc600 Specifies a directory of a "DC600" type streaming tape (implemented only on systems with bus discs having an integral tape backup).

DESCRIPTION

The directory command provides a means of listing the files accessible by the system. Unique qualifiers (File names, user id's, file types, etc.) provide a means of selecting certain files. The special symbols for anycharacter (**[C]** and **[S]**) behave exactly as in the editor string (refer to the Editor Reference Manual for a detailed description).

When a directory command is executed, information regarding the files and bus disc space is given. At the end of each userid is listed the number of files listed for that userid and the total amount of bus disc space (in percent) which that user is occupying. After all userid's are listed, the total amount of bus disc space available is listed. The total space used is the summation of all space used by the various users.

When a recoverable file directory command is executed, an accounting of the total number of files listed is given along with the total amount of bus disc space used. The space used is a percent of the total amount of bus disc space available.

When a directory listing of a multidisc system is run, the listings are in bus disc order—i.e. disc0, disc1, etc.. Each userid has the number of files listed and the percent of bus disc space used. The last listing for the bus disc will include the percent of bus disc space available. The command `directory all_files:HP` will display the system software and date.

SYNTAX

<code>edit [<FILE>]</code>	<code>[</code>	<code>:source</code>	<code>]</code>	<code>[into <FILE>]</code>
		<code>:listing</code>		

Default Values

<FILE> The input file defaults to a new file of type source to be named at completion of edit.

 The destination file defaults to the input file name (if specified). This facilitates editing a file and keeping the edited version by the same name.

`:source` Defaults to type source if neither is specified.
`:listing`

Examples:

`edit` Requests the editor for the purpose of editing a new file to be named at completion of edit.

`edit into PROG1` Requests the editor for the purpose of editing a new file to be named PROG1 with the current userid.

`edit PROG1` Requests the editor for the purpose of editing an already existing source file named PROG1 with current userid.

`edit OUTPUT:listing` Requests the editor for the purpose of viewing a listing output created by a previous list destination specifier of OUTPUT under current userid.

FUNCTION

The edit command evokes the editor to create a new file or edit an existing file. A file may be edited into another file through the use of the key word "into".

Parameters:

<FILE>	Input file is the file to be edited.
	Destination file is where edited file is to be kept upon completion of edit.
:source	Specifies edit of the source file by the given name as opposed to the listing file.
:listing	Specifies the desire to view a listing file by the given name.

DESCRIPTION

The edit command is the user interface to the text editor. It allows specified files to be edited or viewed as well as specification of destination of edited files. Once a file is correctly specified, the editor is entered and, until ended or reset, further commands are for the purpose of text editing. Refer to the Editor Manual for a description of the Editor.

SYNTAX

```
*emulate    [<FILE>] [load <FILE>] [options { edit  
                                           } ]
```

*Appears only when a single emulator/analyzer and no state or timing analysis options are installed.

Default Values

<FILE>

The command file default is to create a new emulator command file.

The load file default is to load no absolute file at time emulation is begun.

Examples:

emulate CMD6800

Requests that the emulator be run and configured with the information CMD6800 type emul__com.

emulate load
PROGABS

Requests the emulator be run with an opportunity to create a new emul__com file describing emulation configuration and that file PROGABS type absolute be loaded into emulation RAM.

emulate CMD8080 options edit

Requests the emulator to be run with an opportunity to edit emul__com file CMD8080 prior to entering emulation. The processor will be restarted.

emulate CMD8080 options continue

Requests the emulator to be entered without restarting the processor. The status of the processor on entry is not necessarily "ready" but rather whatever state the processor is currently in. This allows for exit and resumption of emulation without disturbing the processor for hardware purposes.

FUNCTION

Evokes the emulation function installed in the mainframe. Refer to the appropriate Processor Emulation Manual for a detailed explanation.

Parameters:

- | | |
|-------------|--|
| <FILE> | The command file is of type emul_com and was created in a previous configuration interrogation session in the emulator. It allows the user to evoke the emulator with the same configuration previously specified. |
| load <FILE> | The load file is of type absolute and allows the user to specify that an absolute file created by the linker is to be loaded into emulation RAM. |
| edit | Allows the user to edit responses in an existing emul_com file. |
| continue | Allows the user to enter emulation without restarting the processor. |

DESCRIPTION

The emulate command is a means of accessing the 64000 emulator. The emulator together with hardware options allows execution of an absolute file and optionally I/O simulation. With certain analysis options, real-time analysis of executing programs is possible to aid both hardware and software development. Once the emulate command is correctly specified, further commands are for direction of emulation. Refer to the specific Emulation Reference Manual for a description of the emulator. The (emulate) softkey display on the CRT will vary with the hardware configuration of the system. If a state or timing analyzer is installed in the system, along with the emulation hardware, the softkey display will be (meas_sys). Issuing the "measurement_system" command will allow the user to select (by way of a new softkey configuration) either emulation, state analysis, or timing analysis.

SYNTAX

```
*floppy { utilities  
          system_generator  
          lif_utilities }
```

*Appears only on stations with floppy option installed.

FUNCTION

Allows entry into floppy utilities monitor or floppy system generator monitor.

Parameters:

utilities	Specifies entry into monitor for formatting, testing, and physically duplicating floppies.
system_generator	Specifies entry into monitor for generating floppy discs containing specific system software.
lif_utilities	Specifies entry into the monitor for generating floppy discs in Logical Interchange Format (LIF).

DESCRIPTION

The softkey `(utilities)` accesses the five utility routines: format, copy, verify, read test, and write test. The softkey `(system_generator)` accesses the system software module manager (SSMM). The SSMM allows the user to show, copy, or remove system modules on the bus disc(s) or flexible disc(s). Refer to the Flexible Disc Drive Reference Manual for a detailed description. The softkey `(lif_utilities)` accesses the commands: store, restore, verify, delete, directory format, duplicate, and compare for floppies stored in LIF. Refer to the Floppie Disc Reference Manual for a detailed description.

library

SYNTAX

```
library <FILE> to <FILE>
```

Examples:

```
library OLDRELOC  
to NEWLIB
```

- If NEWLIB:relocatable does not exist it is created and contains identically OLDRELOC.

```
library NEXTRELOC  
to NEWLIB
```

- Now NEXTRELOC is appended to NEWLIB which contains OLDRELOC.

FUNCTION

The library command is used to build libraries of relocatable files for use by the linker. Library files consist of relocatable files to be selectively loaded by the linker.

Parameters:

<FILE> is assumed to be type relocatable.

DESCRIPTION

A library file is merely several relocatable files appended together to form one large relocatable file. Note that the "to" <FILE> need not already exist. If it does exist, the specified file is appended to it. If it does not exist, it is created and filled with the specified file.

Libraries provide a convenient way to specify a group of relocatable files to be used at link time. It is often more efficient for the linker to handle library files since some of the overhead of finding and opening multiple relocatable files is eliminated.

SYNTAX

```
link [<FILE>] [listfile <list destination>]
      [options [edit][nolist][xref][no_overlap_check][comp_db]]
```

Default Values

<FILE>	If no linker command file is specified, the default allows creation of a new file of type link__com.
<list destination>	Defaults to user specified listfile default. See userid command.
options	If "options" is not entered, listing defaults to options specified in the linker command file. If options is specified, followed by nothing, a load map listing with no cross-reference is performed.

Examples:

link	Requests the linker to create a new linker command file. Listing output will go to the listfile default.
link PROGABS	Links absolute file PROGABS containing files in linker command file PROGABS. Listing output will go to listfile default and options in PROGABS type link__com are in effect
link PROGABS options edit	Request the linker for purpose of viewing or modifying PROGABS:link__com. Listing output will go to listfile default.

FUNCTION

The linker combines and relocates specified relocatable files creating an absolute file with the same name as that of the link__com file which can be used to program a PROM (with prom programmer option) or to load emulation RAM to be executed and analyzed with the emulator.

Parameters:

<FILE>	A file of type link__com to be used to direct the linker as to relocatable and relocation addresses.
<list destination>	File or device to which listing output is sent.
options	Allows user to override options specified in the linker command file.
nolist	Overrides the list option specified in the linker command file and suppresses output of a load map.
xref	Overrides no xref option specified in the linker command file and forces output of a global symbol cross-reference table.
edit	Allows user to edit existing link__com file specified.
no__overlap__check	Overrides overlap__check option specified in the linker command file and suppresses errors caused by memory overlaps.
comp__db	This file is created by the linker when requested and is a data base containing information from all of the comp__sym files associated with relocatables in an absolute file.

DESCRIPTION

The link command is the user interface to the 64000 linker module. Once the link command is correctly specified, further responses are used to direct the linker. Refer to the Assembler/Linker Reference Manual for a description of the linker.

SYNTAX

<pre>log_commands { to <FILE> } { off }</pre>

Examples:

log_commands to
COMFILE

After issuing this command, all further commands issued to the system will be written to source file COMFILE for use as a system command file at a later time.

log_commands off

Ends the logging of commands to a source file.

FUNCTION

This command permits the logging of commands to a given file. This allows the generation of complex command files.

Parameters:

<FILE>

Name of source file to which commands are to be logged.

DESCRIPTION

This command permits the building of complex command files. Each command (syntactically correct) that is executed is logged in sequence in the <FILE>. Once the desired commands have been entered, type log_command off in the command line. This terminates the log_command function.

Note: either evoking another command file or editing the file being logged to will implicitly force an end to logging.

measurement_system

SYNTAX

```
*measurement_system [options continue]
```

```
*Appears only when two (2) or more emulators, state  
analyzers, or timing analyzers are installed.
```

Default Values

options	If options continue is not specified, all modules are re-initialized on entry.
---------	--

FUNCTION

Allows the user to select from among emulators, state analyzers, timing analyzers, and software performance analyzers installed and to specify their interaction.

Parameters:

continue	Enter measurement_system monitor, leaving all modules in the state which they were in when last ended.
----------	--

DESCRIPTION

Refer to the State Analyzer, Timing Analyzer, or Software Performance Analyzer Reference Manual for a detailed description of the measurement system monitor.

SYNTAX

```
option_test
```

FUNCTION

The option_test command allows the user to verify the performance of optional boards installed in the system's card cage.

DESCRIPTION

For a detailed description of a specific performance verification refer to the appropriate service manual.

prom_programmer

SYNTAX

```
prom_programmer
```

FUNCTION

Evokes the `prom_programmer` for the purpose of reading from, writing to, or verifying supported programmable read-only memory (PROM) ICs.

DESCRIPTION

Once this command is executed, further commands direct the `prom_programmer` utility. Refer to the PROM Programmer Reference Manual for a description of the `prom-programmer`.

SYNTAX

```
purge <FILE> [:<file type>]
```

Default Values

<file type> Defaults to all types.

Examples:

purge BOB1 Purges all files named BOB1 with the current userid of all types.

purge BOB1:source Purges source file BOB1 with the current userid.

FUNCTION

Remove designated files from active file list.

Parameters:

<FILE> Name of file to be purged.

<file type> Type of file to be purged.

DESCRIPTION

As files are purged, the bus disc space allocated to them is freed for any other file use. The purged file will remain recoverable until the bus disc space is needed for an active file. There is no guarantee on the length of time a file will be recoverable.

The structure of the purge command is such that the exact file(s) must be specified. This prevents accidental purging of files.

recover

SYNTAX

```
recover <FILE> [:<file type>] [to <FILE>]
```

Default Values

<file type>	Defaults to all types.
<FILE>	The destination file defaults to the same name as that of the file to be recovered.

Examples:

recover FILEA	If recoverable and not already existing, all type names FILEA are recovered.
---------------	--

recover FILEA to FILEX	Again, types named FILEA will be recovered if possible and named FILEX.
---------------------------	---

FUNCTION

If a file is recoverable, this command will recover the file and place it on the active file portion of the bus disc.

Parameters:

<FILE>	Name of file to be recovered.
<file type>	Type of file to be recovered.
to <FILE>	New name of file recovered.

DESCRIPTION

When a file is purged, the bus disc space allotted is freed for other files. As more active file space is needed, the purged files are written over. The files are stored under a first purged, first written over philosophy. There is no guarantee as to how long a file is recoverable.

Due to the limited size of the recoverable files directory, the number of recoverable files cannot exceed 127 (63 on floppies). Also, if the space reserved for a recoverable file is needed for an active file the number of recoverable files will be reduced.

The directory list of recoverable files (see directory syntax) provides a list of current recoverable files.

If the file is not available, the system responds with "file not found" and the file must be restored from tape or regenerated.

Another unique feature of the recover command is that a file which has been edited several times may be recovered back to the original source. This is accomplished by recovering the file and renaming it to another file name until all files have been recovered.

Examples:

recover FILEA:source to FILE1

This recovers the most recent edited source.

recover FILEA:source to FILE2

This recovers the next most recent edited source.

recover FILEA:source to FILE3

This recovers the next most recent edited source.

Repeat this command naming a new "to <FILE>" until the desired version is recovered or the system responds with "file not found".

rename

SYNTAX

```
rename <FILE> [:<file type>] [to <FILE>]
```

Default Values

<file type> Defaults to all types

Examples:

```
rename FILEA to FILEB
```

Files names FILEA with the current userid of all types are renamed FILEB.

```
rename FILEA:source to FILEB
```

Source file FILEA with the current userid is renamed FILEB.

FUNCTION

Rename allows the user to change the name of a file(s).

Parameters:

<FILE> Current name of file(s).
to <FILE> New name of file(s).
<file type> Type of file to be renamed.

DESCRIPTION

This function permits the renaming of a file.

SYNTAX

```
*restore [ <file group>[:<file type>][<date qualifier>]
          **file # FILE# into <FILE> [:<file type>]
          ***from_dc600 [ DC600# ] [into_disc DISC#]
                        [ <FILE> ]

          [listfile <list destination>]
```

*Appears only on systems with tape, floppy, or "DC600" type backup running from a bus disc.

**Appears only on systems with tape option.

***Appears only on systems with bus disc having an integral tape backup. NOTE: discs requiring multiple DC600 cartridge for backup do not allow selective file restore. The two-tape pair must be restored completely to the disc.

Default Values

<FILE>	Defaults to all files.
DC600#	Defaults to logical unit 8 (the first integral tape backup on the bus system).
DISC#	Defaults to logical unit 0.
<file group>	Defaults to files of any name with current userid on all bus disc.

<file type>	Defaults to all types. With a file # referenced file, the default is source.
<date qualifier>	If no date qualifier is specified, dates are ignored and do not affect files restored.
<list destination>	Defaults to user specified list file default. See userid command.

Examples:

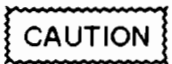
restore Restores from tape or floppy, all files with current userid of all types on all bus discs.

restore FILEA:source Restores from tape or floppy, source file FILEA with current userid, all bus discs.

restore file #2 into FILEB:source
Restores from tape, the second file into a source file named FILEB with the current userid on all bus disc.

restore FILEB:absolute
Restores from tape or floppy, file FILEB, current userid, type absolute on all bus disc.

restore from __dc600 Restores all of "DC600" backup tape. LU8 defaults to tape at LU 0. For 7908, 7911, and 7912 a specific single file may be restored to the disc.



Restoring a full "DC600" tape cartridge completely restores on the disc the information backed up. ALL INFORMATION ON THE DISC IS LOST.

FUNCTION

Restore is used to copy tape or floppy file(s) to the bus disc. Information on tape must be in Standard Interchange Format (SIF). Restore also allows recovery of information stored on "DC600" cartridges. This may be done selectively or the entire "DC600" cartridge contents may be restored, losing the information on the disc.



Parameters:

<FILE>	Name of the file to be restored.
DC600#	Logical unit number of bus disc having an integral tape backup (8-15 allowed).
DISC#	Logical unit number of system bus disc (0-7 allowed).
<file group>	Specifies a file or group of files to be restored.
<file type>	Type of files to be restored.
<date qualifier>	Allows restoration of files selectively based on date of last modify or access.
FILE#	Sequence number of file as stored on tape. May be listed with directory command for tapes generated on the 64000.
<list destination>	Specifies file or device to receive listing output.

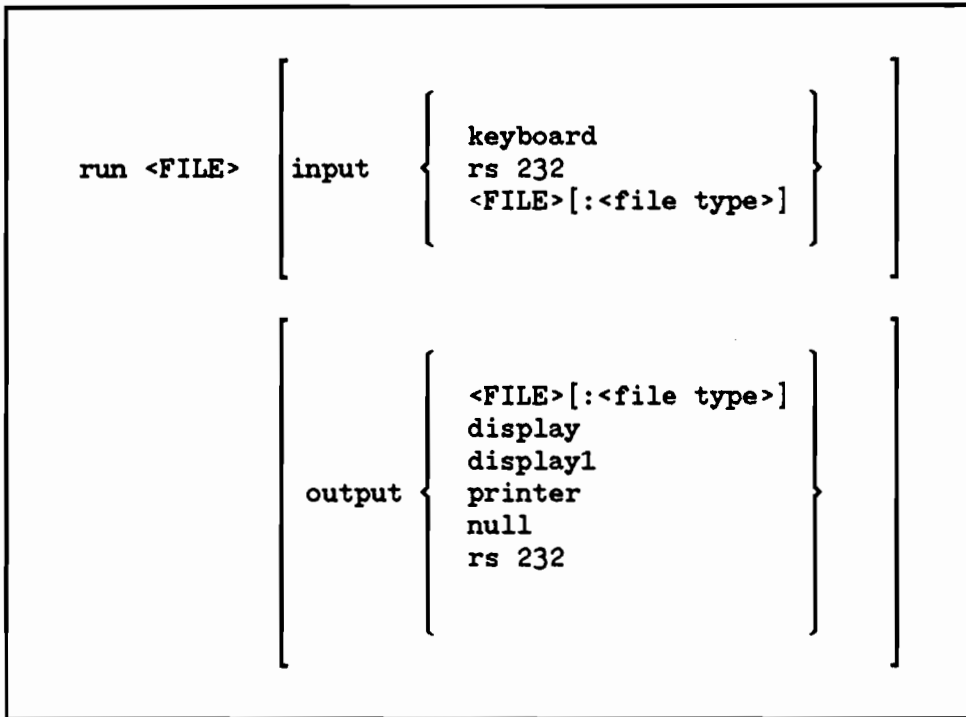
DESCRIPTION

The restore command allows the user to retrieve "backed-up" files from tape or floppy. Files stored on tape with another system (provided that system used Standard Interchange Format (SIF) in creating that tape) may be restored on the system. For files stored to tape or floppy from the 64000, retrieval may be accomplished by restoring the file by name. For files stored (using SIF) by another system, files may only be restored by file # reference since the directory cannot be read or may not be present. In the latter case, a file name must be given to the file for purposes of storage on bus disc. Such file # referenced files are assumed to be of type source. It is very unlikely that files of other types adhere to the 64000 conventions.

Users with bus discs having the integral tape backup may restore from backup (if disc has been stored to backup). This may be done altogether (losing disc information) or by filename. Users with floppy drive may restore files stored by the 64000 either by name or by using the all_files and all_userids commands.

run

SYNTAX



Default Value

input if the input keyword is not used, then the file name associated with INPUT is keyboard.

output if the output keyword is not used, then the default file name for OUTPUT is the default listing file. If no default listing file has been specified for the present userid, then the default listing file is null.

<file type> If no file type is specified, the input default is to source and the output default is to listing.

FUNCTION

The run command causes execution of an absolute pseudo-machine code file that has been compiled on the system Host Pascal compiler.

Parameters:

<FILE>	The name of the absolute file containing the pseudo-machine code.
input <FILE>	Disc file from which predefined file INPUT gets data.
Keyboard	Specifies input from keyboard is to be read as records when predefined file INPUT is accessed.
output <FILE>	Disc file to which predefined file OUTPUT writes data.
<file type>	Type of file from which input is read or to which output is written.
display	When OUTPUT is written to, records scroll past on display.
display 1	Allows random access writes (row and column) to display using predefined file OUTPUT.
printer	When OUTPUT is written to, records go to printer.
null	When OUTPUT is written to, records are not saved.
rs 232	Capability to input/output over RS-232-C.

DESCRIPTION

File names for the standard files INPUT and OUTPUT may be specified in the context of the run command. For other program parameters, the system prompts the user for a file name for each additional program parameter. If no entry is made following a prompt, the system treats the file as if it were not a program parameter. For additional information covering the run command refer to the 64000 Host Pascal Reference Manual.

state

SYNTAX

```
*state [<FILE>][options continued]
```

*Appears only when a single state analyzer and no emulator, timing analyzer, or software performance analyzer boards are installed.

Default Values

<FILE>	If no trace file is specified, entry is such that configuration information from previous use of state analysis is not used.
options	If options continue is not specified, default is to re-initialize the state module upon entry.

Example:

state CONF 1	Enter state analysis operation using configuration information contained in file CONF1:trace. Re-initialize state upon reentry.
state options continue	Enter state analysis operation using configuration contained in the configuration file used the last time state was entered. Does not re-initialize state upon entry.

FUNCTION

Evokes the state analysis function installed in the mainframe.

Parameters:

<FILE>	Name of trace file containing state configuration information.
continue	Re-enter state module leaving it in the same condition as it was when it was last "ended".

DESCRIPTION

The `(state)` softkey displayed on the CRT will vary with the hardware configuration of the system. If an emulator, timing analyzer, or software performance analyzer is installed in the system, along with the state analyzer hardware, the softkey display will be `(meas_sys)`.

Issuing the "measurement_system" command will allow the user to select (by way of a new softkey configuration) either emulation, state analysis, or timing analysis.

Refer to the State Analyzer Reference Manual for a detailed description.

store

SYNTAX

```

**store [ <file group>[:<file type>][<date qualifier>]
        *to_dc600[DC600#][from_disc DISC#][certify]

        [listfile <list destination>]

```

*Appears only on systems with bus discs having an integral tape backup.

**Appears only on stations with tape option, floppy option, or with bus discs having an integral tape backup.

Default Values

<DISC#>	Defaults to logical unit 0.
<file group>	Defaults to files of any name with current userid on all bus disc.
<file type>	If no file type is specified, the types source, emul__com, link__com, and trace are assumed.
<date qualifier>	If no date qualifier is specified, dates are ignored and do not affect files stored to tape or floppy.
<list destination>	Defaults to user specified listfile default.
DC600#	Defaults to logical unit 8 (the first bus disc having an integral tape backup).
certify	If not specified, no certification is performed unless DC600 has never been certified.

Examples:

store Stores (overwriting previous contents) all files of type source, emul__com, link__com, and trace to tape or floppy.

store X:all userids:source

Stores all files named X with any userid on any disc of type source to tape or floppy.

store all__files modified after 15/4/81

Stores all files on any disc of type source, emul__com, link__com, and trace with current userid modified after 15 April, 1981 to tape or floppy.

store all__files:1:DON

Stores all files of type source, emul__com, link__com, and trace from disc # 1 with userid DON to tape or floppy.

store to__dc600

Store entire system disc to "DC600" streaming backup tape at LU 8.

FUNCTION

Store is used to copy bus disc file(s) to tape, floppy, or DC600 backup tape. A directory is placed on the medium written to.

Parameters:

- | | |
|--------------------|--|
| <file group> | Files to be stored on tape or floppy. |
| <file type> | Type of files to be stored on tape or floppy. |
| <date qualifier> | Allows selection of files to be stored based on last modification or access. |
| <list destination> | File or device to which listing output is sent. |
| DC600 | Logical unit number of bus disc having an integral tape backup (8-15 allowed). |
| DISC# | Logical unit number of disc to be backed up. |
| certify | Forces a certification of the "DC600" streaming tape prior to backup. If tape has never been certified previously, certification is automatically performed. |

DESCRIPTION

After files are stored, a directory of files is also placed on the storage media.

To facilitate a regular "back-up" of files on bus disc, it is possible to store to tape or floppy, only those files modified (or accessed) since the last back-up. The third example above shows a command for storing only those files modified after a given date.

When storing to "DC600" streaming tapes, all of the information on the disc is stored to the tape including the disc directory. Subsequently, it is possible to restore selectively specific files from the "DC600" tape or to restore the entire disc to its state at the time the backup was made.

"DC600" streaming tapes must be certified before first used. Thereafter, the user may force a "certify" using the certify parameter. Following the store of data, a checksum verify is automatically performed on the tape data.

SYNTAX

`terminal`

FUNCTION

Executes terminal mode software to allow communication with other systems over RS-232-C.

DESCRIPTION

Once this command is executed, further commands direct the terminal mode utility. Refer to the Terminal Mode Operating Manual for a detailed description of the terminal mode of operation.

timing

SYNTAX

```
*timing [<FILE>][options continue]
```

*Appears only when a single timing analyzer and no emulator, state analyzer, or software performance analyzer boards are installed.

Default Value

<FILE>	If no trace file is specified, entry is such that the configuration information from previous use of timing analysis is not used.
options	If options continue is not specified, default is to re-initialize the timing module upon entry.

Example:

timing CONF 2	Enter timing analysis operation using configuration information contained in file CONF2:trace and re-initialize timing upon entry.
timing options__continue	Enter timing analysis operation using configuration data contained in the configuration file that was used the last time the timing command was entered. Does not re-initialize timing upon entry.

FUNCTION

Evokes the timing analysis function installed in the mainframe.

Parameters:

<FILE>	Name of trace file containing timing configuration information.
continue	Re-enter timing module leaving it in the same condition as it was when last "ended".

DESCRIPTION

The **(timing)** softkey display on the CRT will vary with the hardware configuration of the system. If an emulator, state analyzer, or software performance analyzer is installed in the system, along with the timing analyzer hardware, the softkey display will be **(meas_sys)**. Issuing the "measurement__system" command will allow the user to select (by way of a new softkey configuration) either emulation, state analysis, or timing analysis for system operation.

Refer to the Timing Analyzer Reference Manual for a detailed description.

userid

SYNTAX

```
userid <USERID> [listfile default <list destination>]
                [password <PASSWORD>*]
                *not printed
```

Default Values

<USERID>	If no userid is entered, the "blank userid" is assumed.
<list destination>	If no listfile default is specified, the default is null.
<PASSWORD>	If no password is specified, the userid is not protected.

Examples:

userid	Sets the userid to be blank and the listfile default to null.
userid MAX listfile default printer	Sets the userid to MAX and the listfile default to the printer.
userid MAX password secret	Sets userid to protected userid pMAX provided "secret" is the password or if this is the first time this password has been used, it sets a protected userid pMAX.

FUNCTION

Allows user to select current userid and where listings should be sent by default.

Parameters:

<USERID>	Name of userid to become current userid.
<list destination>	File or device to which listing output is to be sent by default.
<PASSWORD>	Up to 15 characters including all ASCII characters (i.e., alphanumerics, punctuations, and control characters) except spaces.

DESCRIPTION

For purposes of file management, it is convenient to have userid associated with disc files. Since a user will most frequently be referencing files under his own userid, this command allows the user to specify his id which, thereafter, will automatically be assumed for all file references. When a userid is entered, if a source file named WELCOME exists under the new userid, the first 18 lines of file WELCOME are copied to the display. This feature allows users to display information or messages about their files.

Protected userids are not accessible from any userid other than the protected userid itself. Protected userid files will be displayed with other files in directory commands with the "all_userid" specifier when issued from any userid; however, they may only be accessed after the default userid has been set to the protected userid. It should be noted that since the lower-case "p" precedes the specified userid name in a protected userid, only five (5) characters are significant in a protected userid name.

CAUTION

Once entered, the password on a protected userid can only be identified by the system manager (see Chapter 2 of this manual) and the user. The password may however, be changed while operating under that protected userid by re-entering the userid command and changing to a new password.

verify

SYNTAX

```
*verify [ <file group>[:<file type>][<date qualifier>]
        **file # <FILE #> with <FILE> [:<file type>]
        [list <list destination>]
```

*Appears only on systems with tape or floppy backup running from hard disc.

**Appears only on systems with tape option.

Default Values

<file group>	Defaults to files of any name with current userid.
<file type>	Defaults to all types. With a file # referenced file, the default is source.
<date qualifier>	If no date qualifier is specified, dates are ignored and do not affect files compared.

Examples:

verify	Compares with files on bus disc, all tape or floppy files with current userid of all types from all bus discs.
verify FILEA:source	Compares with file on bus disc, source file FILEA with current userid on tape or floppy from all bus discs.

verify file #2 with FILEB

Compares the second file on tape or floppy with FILEB type source with current userid on any bus disc.

FUNCTION

Allows user to verify that a file on bus disc is the same as a file on tape or floppy. Allows the user to verify that data stored on "DC600" tape is readable.

Parameters:

<file group>	Specifies a file or group of files to be verified.
<file type>	Type of file to be verified.
<date qualifier>	Allows verification of files selectively based on date of last modification or access.
<FILE #>	Sequence number of file as stored on tape.
<list destination>	Specifies file or device to receive listing output.

DESCRIPTION

The verify command accomplishes two things: 1) It verifies that the tape or floppy is readable. 2) It verifies the data stored on the tape or floppy against the data stored on the bus disc.

If a "low threshold" message is present during a verify operation, the information on the tape is of marginal quality. An effort should be made to regenerate the tape.

NOTES

Chapter 4

SOFTWARE UPDATE PROCEDURES

SOFTWARE LOADING AND UPDATING PROCEDURE USING FLEXIBLE DISC DRIVE

This procedure should be used to load or update system software after user-generated files have been stored on the bus disc. To initialize, perform performance verification, or fix soft data errors on the bus disc, refer to the Service Overview Manual.

Procedure:



NOTE

In a cluster 64000 installation, this procedure must be run on the master controller; all other stations should be powered off.

- a. Set System Control Source switches to "LOCAL MASS STORAGE", "(LMS) ADRSABLE", or "(LMS) TALK ONLY.
- b. Check software to be booted. If it does not include OP SYSTEM 1 of __ flexible disc, insert original OP SYSTEM #1 disc into flexible disc drive.

NOTE

When loading or updating software from a flexible disc(s), flexible disc OP System #1 must be loaded first. Additional discs should then be loaded in numerical order.

- c. With development station power on, hold **(SHIFT)** key down and press **(RESET)** key once. Disc light should come on and display should indicate *BOOT IN PROGRESS*; if not, cycle station power OFF and then back ON. When the SYSTEM #1 flexible disc is booted, the soft keys appear as follows:

(format) (test) (fmt&fst) (soft_fix) (diag) (menu) () (end)

Press (end) and continue.

- d. Insert other flexible disc in disc drive; disc should begin booting automatically and *BOOT IN PROGRESS* message should appear on the CRT. (Repeat this for each additional disc.)
- e. When all flexible disc loading is complete, remove disc and reset system Control Source Switches to SYSTEM BUS; normal operation is now automatically resumed.

NOTE

For format, test, soft fix, and diagnostic options, refer to the System Diagnostic Manual.

SOFTWARE LOADING AND UPDATING PROCEDURE USING TAPE CARTRIDGE

NOTE

The following procedure is for users that have older Model 64000s with tape cartridge drive capabilities. This capability is no longer supported on the later models.

This procedure should be used to load or update system software after user-generated files have been stored on the bus disc. To initialize, perform performance verification, or fix soft data errors on the disc, refer to the Installation and Configuration Manual.

Procedure:

NOTE

In a 64000 cluster installation, this procedure must be run on the master controller; all other stations should be powered off.

- a. Set System Control Source switches to "LOCAL MASS STORAGE", "(LMS) ADRSABLE", or "(LMS) TALK ONLY".
- b. Check software to be booted. If it includes an Operating System #1 tape, do **NOT** boot tape #1 as it is used only for disc reformatting, performance verification, and soft data error correction. If this tape is booted accidentally the softkeys appear as follows:

(format) (test) (fmt&tst) (soft_fix) (diag) (menu) () (end)

Just press (end) and continue by inserting another tape.

- c. Insert Operating System (or other) tape in front-panel tape drive slot of the master controller station.
- d. With development system power on, hold (SHIFT) key down and press (RESET) key once. Tape light should come on and display should indicate *BOOT IN PROGRESS*; if not, cycle station power OFF and then back ON.
- e. Insert other tape (if provided) in tape drive; tape should begin booting automatically and *BOOT IN PROGRESS* message should appear on the CRT. (Repeat this step for each additional tape.)
- f. When all tape loading is complete, remove tape cartridge and reset System Control Source Switches to SYSTEM BUS; normal operation is now automatically resumed.

NOTE

For format, test, soft fix, or diagnostic options, refer to the System Diagnostic Manual.

NOTES

Appendix A

STATUS AND ERROR MESSAGES

This appendix may be used as a quick reference guide for general status/error messages associated with the system monitor. These messages are not all inclusive. For status/error messages associated with a specific system module (i.e., editor, linker, assembler, etc.) refer to the applicable module reference manual.

The monitor status/error messages are listed alphabetically in two separate tables for convenience of the user.

Table A-1. General Status Messages

Status Messages	Meaning
Awaiting command userid: XXXXXX	System is ready for commands and the default userid is XXXXXX.
CAPS LOCK off	All alphabetic keys default to lower case.
CAPS LOCK on	All alphabetic keys default to upper case.
Cartridge tension complete	Tensioning tape finished.
Commands to file FILE until "log off"	Log commands activated to file.
Copying	One file is being copied to another.
Copy complete, N records transferred	Copy using RS-232-C complete.
Copy complete, N records transferred (rs232 timed out)	Copy using RS-232-C ended by time out on received data.
End of directory, N files listed	Directory is complete.
Fri, 27 Feb, 11:40 (example)	Date or time command has displayed current system date and time.

Table A-1. General Status Messages (Cont'd)

Status Messages	Meaning
INSERT CHAR off	Normal noninserting operation resumed.
INSERT CHAR on	Characters typed will be inserted at special (^) cursor.
I/O Bus reset	The master controller station displays this message when it has had to gain access to a network resource.
I/O Bus wait	The 64000 station displaying this message is waiting for a network resource.
Listing files	Directory is listing files to listfile.
Listing recoverable files	Directory is listing recoverable files to listfile.
No action taken	Log or command copy was aborted by user's choice not to purge existing file.
PAUSED (to initiate RESET, press RESET again)	The RESET key has been pressed once. Pressing any other key resumes normal operation.
Premarking directory	Store command is premarking selected directory.
Reading directory	Selected command is reading media directory.
Recall buffer empty	No valid commands have been issued since boot-up.
Receiving record N	Record N of a file is received over RS-232-C via copy.
Restore complete, files restored=N of M	N of the M files meeting the qualifications given have been restored (for reason others were not restored, see listing).
Rewinding	Tape cartridge is being rewound.

Table A-1. General Status Messages (Cont'd)

Status Messages	Meaning
Searching	Restore or verify is finding file on storage media.
Searching for files	Directory is searching disc for files.
Searching for recoverable files	Directory is reading recoverable file list.
Searching for userids	Directory is scanning disc for all userids.
Store complete, files stored = N of M	N of the M files meeting the qualifications given have been stored (for reason others were not stored, see listing).
Storing FILE	File is being written to storage media.
Tensioning tape	Tension command is tensioning tape.
Transmitting record N	Record N of a file is being transmitted out over RS-232-C via copy.
Verify complete, files verified = N of M	N of the M files meeting the qualifications given have been verified (for reason others were not verified, see listing).
Verifying FILE	File on storage media is being compared to disc.
Waiting for data	Copy from RS-232-C is waiting for data to be received.
Waiting for input	Restore is waiting for response to question.
Waiting for printer	The station displaying this message is waiting for another station which is using the printer.
Writing directory	Station command is updating storage media directory.

Table A-2. General Error Messages

Error Messages	Meaning	Corrective Action
Cartridge out	No data cartridge was detected following a tape command.	Insert data cartridge and reissue command.
Device timeout	Storage media drive did not respond within correct interval.	Check all cable connections to media drive and media control board. If problem continues, contact HP Service Representative.
Directory full: file=FILE:type	File cannot be created --- directory full.	Purge unwanted files.
Disc full - file=FILE:type	Disc is too full to work with named file.	Purge unwanted files.
Disc N down	Disc with LU N is in one of the following states: Power off; Drive fault; Heads not loaded; Busy; Read has timed out.	Check disc LU N for error indications.
Disc number is invalid	An attempt was made to read the directory on a disc whose LU was not defined at power-up.	If the disc was attached after power-up, reboot the system; otherwise, correct the command to a valid LU.
End of file: file=FILE:type	Premature end of file was encountered.	Regenerate file in question.
File exists: file=FILE:type	This error implies attempt to create file by same name.	Purge file, if unwanted, or create a new file.
File may not be appended to itself	A library command has same file as source and destination.	Revise command to use two distinct file names.
File name:HP not present	System software by this name is not on disc LU 0.	Boot from appropriate system software media.

Table A-2. General Error Messages (Cont'd)

Error Messages	Meaning	Corrective Action
File not found	A file in the issued file management command does not exist.	Correct to an existing file name.
File too big for tape, split file into 2 pieces	File with data length greater than tape capacity cannot be stored as a single file.	Modify file so it will fit (probably by splitting a source or by listing in half).
Framing error	During copy from RS-232, one of the following errors has occurred: Rec/trans baud rates are different; Parity error; Character length; Number of stop bits wrong.	Check for consistency in setup of RS232 transmitter and receiver.
Illegal directory or tape format	The data cartridge being read was created on another system which did one of the following: 1. Did not adhere to Standard Interchange Format (SIF) used by 64000 system tape software. 2. Used no directory or directory format different from that of the 64000 tapes.	The information is obtainable from tape only if the tape is in SIF format. To avoid an incompatible directory format, files must be referenced by file number (see restore and verify command syntax).
Illegal disc: file=FILE:type	There is no disc with LU # on this file.	Attach appropriate disc, or reference proper LU #.
Invalid date	An illegal date has been specified.	Revise date per Day/Mon/Year format.

Table A-2. General Error Messages (Cont'd)

Error Messages	Meaning	Corrective Action
Invalid disc number	A disc LU number not assigned was used.	Revise LU number to one currently assigned.
Invalid input, question must be answered	A nonreproducible file which already exists is about to be overwritten and user permission is being requested.	Answer question: "yes" - I do mean to overwrite the file; or "no" - I do not want the file overwritten.
Invalid time	An illegal time has been specified.	Revise time per HH:MM format.
Invalid to recover across discs	A recovery has been attempted from one LU to another.	Revise command to use only one LU.
Invalid to rename across discs	A name has been attempted from one LU to another.	Revise command to use only one LU.
"log_commands" not permitted in command file	A "log_commands" command was in cmdfile.	Remove "log_commands" from command file.
Maximum parameter name size exceeded	During read of PARMs statement in a command file, the maximum number of characters in parameter name was exceeded.	Use shorter or fewer parameter names.
Maximum parameter text length exceeded	The absolute maximum length of text to be substituted for cmdfile parameters has been exceeded.	Shorten number of ASCII characters in text being passed to cmdfile parameters.
More parameters passed than declared	During invocation of command file, text for more parameters than declared was passed.	Declare additional parameter or correct text passed.
Parameter &NAME not found	During execution of a command file, an undeclared parameter was found	Examine command file and correct parameter name or declaration.

Table A-2. General Error Messages (Cont'd)

Error Messages	Meaning	Corrective Action
Printer down	Printer is powered off or is off line.	Check printer.
Powerfail or bus reset	During tape or floppy operation, a power failure or bus reset occurred, aborting the operation.	Repeat the attempted operation.
Recovered	System has recovered from one of the last three conditions.	None.
Retrying disc N	One of the following has occurred accessing disc LU N: Seek check; Head just reloaded; Transmission error; Cylinder, Head, Track compare error; Data error; Illegal track access; Head off track; Write to protected track.	Wait for retry-in to recover itself. If it does not, reboot system and disc. If trouble continues, call HP Service Representative.
Servo Failure	A servo failure in the storage media drive was detected.	Check all cable connections to media drive and media control board. If problem continues, contact HP Service Representative.
Syntax error	Command just issued has a syntax error at the point where cursor is located.	The softkeys are labelled to indicate the valid syntax at the cursor position; modify command to reflect a valid choice.
Syntax error S1	The system has run out of available memory.	Reboot; if problem occurs frequently, call HP Service Representative.

Table A-2. General Error Messages (Cont'd)

Error Messages	Meanings	Corrective Action
Tape cartridge hardware not present	No tape control board was detected in the card cage of station on which a tape command was given.	Check connection or tape control board, if present. Issue command on station with tape hardware.
Tape's directory is empty	An attempt was made to read the directory of a tape which is premarked but has no files stored on it.	Tape is empty and should now be accessed only by a store command.
Write protected	A command requesting a write of data to the storage media was issued and the write protect tab is in the protect position.	To write to the media, remove the write inhibit tab.

Index

The following index lists important terms and concepts of this manual along with location(s) in which they can be found. The numbers to the right of the listings indicate the following manual areas:

- **Chapters** - references to chapters appear as "Chapter X", where "X" represents the chapter number.
- **Appendixes** - references to appendixes appear as "Appdx Y", where "Y" represents the letter designator of the appendix.
- **Figures** - references to figures are represented by the capital letter "F" followed by the figure number.
- **Other entries** - reference to other entries in the index are by page number.

a

absolute file type	2-1
append Syntax	3-14
asmb__db file type	2-1
asmb__sym file type	2-1
assemble Syntax	3-16

c

<CMDFILE> Syntax	3-18
Commands:	
Editing	1-4
Entering	1-3
Logging	1-9
Recalling	1-4
Re-executing	1-4
Stopping Execution	1-5
Command Files	1-6
Command File Structure	1-7
comp__db file type	2-1
comp__sym file type	2-1
compile Syntax	3-20
copy Syntax	3-22

d

data file type	2-1
<DATE> Syntax	3-5
<date qualifier> Syntax	3-6
date&time Syntax	3-24
directory Syntax	3-26
Directory Format	2-4
Display Format	F1-1

e

edit Syntax	3-29
emulate Syntax	3-31
emul_com file type	2-2
Error Messages	Appdx A

f



<FILE> Syntax	3-7
<file group> Syntax	3-9
<file type> Syntax	3-10
Files, type	2-1
File Manager	2-2
File Security	2-3
Flexible Disc Utility	4-1
floppy Syntax	3-33

l

library Syntax	3-34
Libraries, user	2-4
link Syntax	3-35
link_com file type	2-2
link_sym file type	2-2
<list destination> Syntax	3-11
listing file type	2-2
log_commands Syntax	3-37

m

measurement_sys Syntax	3-38
Messages	
Error	Appdx A
Status	Appdx A

o

option_test Syntax	3-39
---------------------------------	------

p

Page Structure	2-4
Parameter passing	1-6
Predefined Syntax Elements	
<DATE>	3-5
<date qualifier>	3-6
<FILE>	3-7
<file group>	3-9
<file type>	3-10
<list destination>	3-11
<USERID>	3-12
prom file type	2-2
prom_programmer Syntax	3-40
purge Syntax	3-41

r

recover Syntax	3-42
reloc file type	2-2
rename Syntax	3-44
restore Syntax	3-45
run Syntax	3-48

s

Software Loading	
Flexible Disc	4-1
Tape	4-2
state Syntax	3-50
Status Messages	Appdx A
store Syntax	3-52
System file type	2-2
System monitor	3-1

t

TAPE Utility	4-2
terminal Syntax	3-55
time Syntax	3-56
trace file type	2-2

u

userid Syntax	3-58
<USERID> Syntax	3-12

v

verify Syntax	3-60
----------------------------	------

NOTES



64980-90934, APRIL 1984
Replaces: 64980-90928, January 1983



PRINTED IN U.S.A.