HEWLETT PACKARD

General User's Reference Manual
MPE V



HP 3000 Computer Systems

MPE V

General User's Reference Manual



Edition 1 E1088

32033-90158
Printed in U.S.A. 10/88

Notice

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

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

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

This document contains propriety information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company.

© Copyright 1988 by HEWLETT-PACKARD COMPANY

Printing History

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

The software code printed alongside the date indicates the version level of the software product at the time the manual edition or update was issued. Many product updates and fixes do not require manual changes, and conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

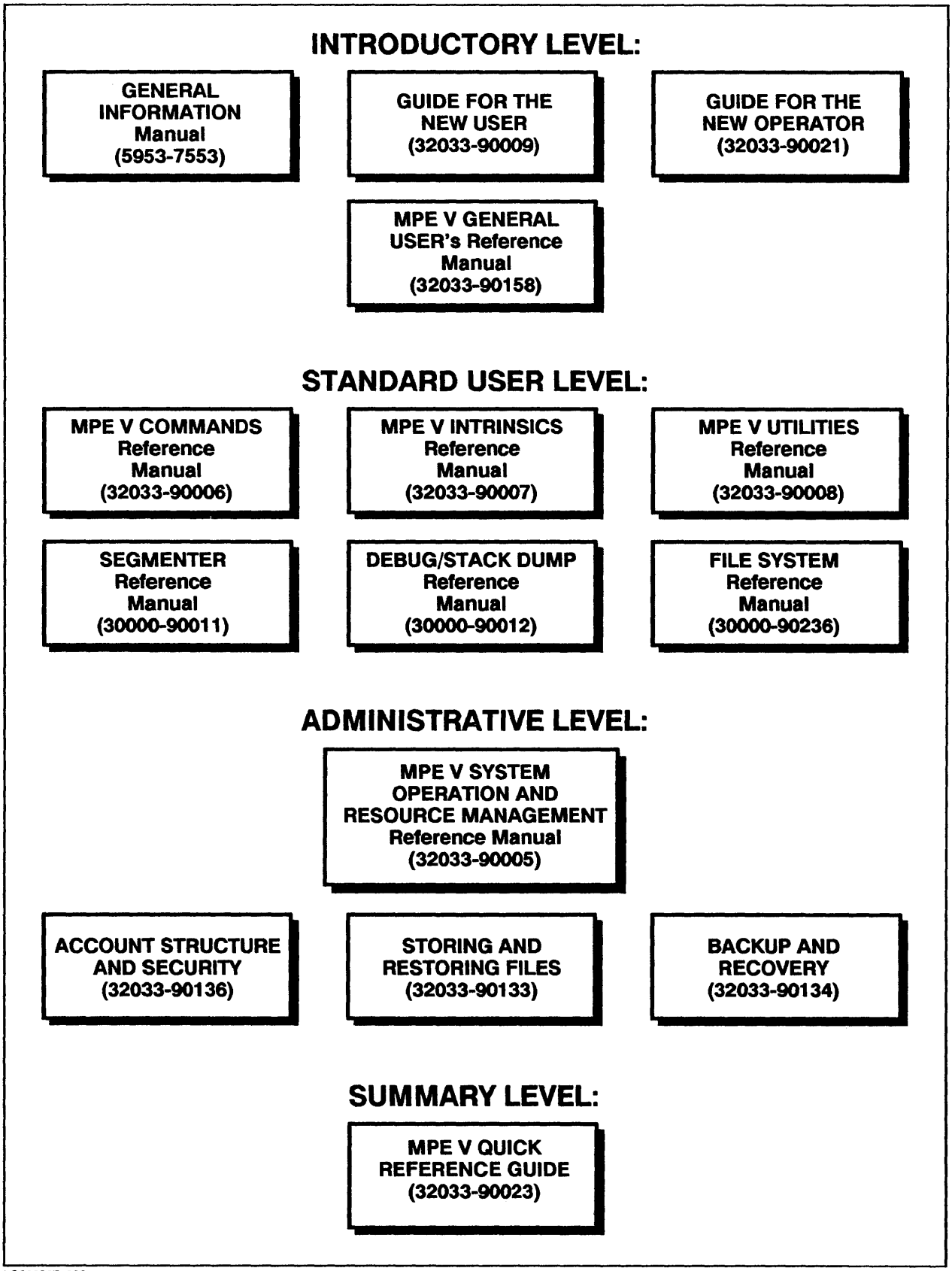
First Edition . . . . . .October 1988

List of Effective Pages

The List of Effective Pages gives the date of the most recent version of each page of the manual. To verify that your manual contains the most current information, check the dates printed at the bottom of each page with those listed below. The date on the bottom of each page reflects the edition or subsequent update in which that page was printed.

Effective Page	Date
All	October 1988

MPE V Manual Plan



LG200027_009

Preface

Introduction

Welcome to the *MPE V General User's Reference Manual* for the HP 3000's MPE V (Multi-Programming Executive) operating system.

As the title implies, this manual was designed and written for general users. This includes anyone who wants to know more than how to log on and work with a specific application. That may include clerical personnel, managers, and other professionals who use MPE to do their work. It also includes the technical personnel who run and maintain the system.

The manual assumes minimal prior experience with computers. However, it is not designed as a training tool to help novice computer users to get started and become productive. Instead, it provides general users with more complete information once they have some experience with the system.

NOTE

For self-paced training at the beginning level, refer to *Guide for the New User* (32033-90009).

Objectives

Specifically, this manual has the following objectives:

- To provide an intermediate level of information that allows users to understand and use more advanced manuals and techniques.
- To provide concise, step-by-step procedures for tasks related to the manual's topics.
- To serve as an introduction and orientation to technical personnel, who need to know how to apply their skills and knowledge to the HP 3000.

How to Use this Manual

This manual was not designed to be read from beginning to end. Instead, users should consult this manual for additional information on a given topic or to find out how some piece of information fits into the larger picture.

NOTE

We recommend reading Chapter 1 first, especially for novice users. All subsequent chapters assume familiarity with the concepts in this chapter.

Manual Organization

This manual is organized into chapters according to major topics. Each chapter also contains references to other manuals that relate to those topics. In addition, there are several appendices that are not necessary for all readers, but may be useful or of interest to some.

Some chapters also contain streamlined procedures, such as instructions for displaying files on the terminal. Note that procedures are listed with the topics to which they relate, which is not necessarily the order in which they are performed. To help find procedures quickly, this manual provides a separate table of contents for procedures.

Chapter 1 **Basic Concepts** provides a system overview and describes basic concepts.

Chapter 2 **The Terminal** describes how to use the terminal.

Chapter 3 **Introduction to Commands** provides an overview of command components.

Chapter 4 **UDCs** describes shortcuts to commands.

Chapter 5 **Utilities and Subsystems** describes how to use the HP 3000's subsystems.

Chapter 6 **Basic Account Structure** describes how information is organized and protected.

Chapter 7 **Working with Files** provides information for creating and working with files.

Chapter 8 **Working in Sessions** describes how to work in interactive sessions.

Chapter 9 **Jobs** describes how to set up and run batch jobs.

Appendix A **Summary of Commands that Execute "In Break"**

Appendix B **Procedures for using the DSCOPY Subsytem**

Table of Contents

Chapter 1
Basic Concepts

Chapter Overview 1-1
Types of Computers 1-2
  Microcomputers 1-2
  Minicomputers 1-3
  Mainframe Computers 1-3
Hardware Components of a Computer System 1-4
  The HP 3000 Computer 1-4
  Peripherals 1-5
    Terminals 1-5
    Disc Drives 1-5
    Tape Drives 1-6
    Printers 1-7
Software Components of a Computer System 1-8
  MPE Operating System 1-9
  Information Management Systems 1-9
  The Subsystems 1-9
Overview: Working with the HP 3000 1-10
  Sessions and Jobs 1-10
  Defaults 1-11

Chapter 2
The Terminal

Chapter Overview 2-1
Introduction 2-2
Things to Check 2-2
  Connections 2-2
    Power Cable Connection 2-3
    Computer Cable Connection 2-3
    Keyboard Cable Connection 2-5
  Terminal Settings 2-5
    Remote Mode (ON) 2-6
    Block Mode (OFF) 2-6
    Auto LF (OFF) 2-6
    Baud Rate 2-7
    Parity (0 or NONE) 2-7
    Enq/Ack (YES) 2-7
  Checking and Changing Terminal Settings 2-7
    Checking Remote Mode, Block Mode, and Auto LF 2-8
    Checking the Baud Rate, Parity, and Enq/Ack Settings 2-9
The Keyboard 2-11
  Scrolling 2-11
  Comparison with Typewriter Keyboards 2-12
  The "Special" Keys 2-14
    The Terminal Control Keys 2-14
    The Edit Control Keys 2-15
    The Cursor Control Keys 2-15
    The Command Keys 2-16
    The Function Keys 2-18

Table of Contents (Continued)

Printing What Is on the Screen 2-21
Printing Screen Dumps at Your Workstation 2-21
Printing Listings on the System Printer 2-23

Chapter 3
Introduction to Commands

Chapter Overview 3-1
Introduction 3-2
Anatomy of a Command 3-2
Prompt 3-3
Command Name 3-3
Parameters 3-4
Required Parameters 3-4
Optional Parameters 3-4
Positional and Keyword Parameters 3-5
Summary: Characteristics of Parameters 3-6
Delimiters 3-6
The Ampersand Continuation Character 3-7
Reading Documentation 3-8
Reading the Commands Reference Manual 3-8
Command Name 3-8
Syntax 3-8
Parameters 3-9
Use 3-9
Operation 3-10
Examples 3-10
Additional Discussion 3-10
Summary: Reading the Commands Reference Manual 3-10
Reading the Quick Reference Guide 3-11
The HELP Facility 3-12
Using the HELP Facility 3-12
Finding General Information about Commands 3-12
Finding Specific Information about Commands 3-14
Overview of Common Commands 3-17
Monitoring Commands 3-17
File Commands 3-18
Session Commands 3-19
Job Commands 3-20
Message Commands 3-21

Chapter 4
UDCs

Chapter Overview 4-1
Introduction 4-2
UDCs and Function Keys 4-2
UDC Levels 4-2
UDC Construction 4-3
Header 4-3
UDC Name 4-4
UDC Parameters 4-4

Table of Contents (Continued)

Execution Options 4-4
LIST/NOLIST 4-4
BREAK/NOBREAK 4-4
LOGON/NOLOGON 4-5
HELP/NOHELP 4-5
Body 4-5
End 4-5
Creating UDCs 4-6
Creating Nested UDCs 4-8
Adding, Modifying and Removing UDCs 4-10
Finding out about UDCs on the System 4-16
Using the :SHOWCATALOG Command 4-16
Using the HELP Facility with UDCs 4-17

Chapter 5
Utilities and Subsystems

Chapter Overview 5-1
Introduction 5-2
EDITOR 5-4
About Line Numbers 5-4
Specifying Line Numbers 5-4
Saving Line Numbers 5-5
About Scratch Files 5-5
Using EDITOR 5-5
Creating a New File 5-6
Modifying an Existing File 5-9
Inserting Text into Existing Files 5-11
Moving Lines within the File 5-13
Copying Lines within the File 5-14
Deleting Lines within a File 5-16
Printing Files 5-19
Using EDITOR's HELP Facility 5-20
SORT-MERGE 5-20
Required File Format 5-21
Creating Files to be Sorted and Merged 5-21
Using the SORT Program to Sort Files 5-23
Displaying or Editing a Sorted File 5-25
Using the MERGE Program to Combine Sorted Files 5-26
STORE/RESTORE 5-27
About the Tape 5-28
Using the STORE Subsystem 5-29
The Execution Options 5-30
The SHOW Parameter 5-30
The PURGE Parameter 5-31
The PROGRESS Parameter 5-32
Using the RESTORE Subsystem 5-32
FCOPY 5-34
Using the FCOPY Subsystem 5-35
Copying Files between Accounts 5-36
Comparing File Contents with FCOPY 5-38
Printing Files with the FCOPY Subsystem 5-38

Table of Contents (Continued)

SPOOK 5-39
About Spoolfiles 5-39
About SPOOK Tapes 5-39
The SPOOK Commands 5-39
Using the SPOOK Subsystem 5-40
Getting Started 5-40
Displaying Spoolfiles 5-41
Locating Data in Spoolfiles 5-42
Adding to (Appending) Spoolfiles 5-43
Deleting Spoolfiles 5-44
Changing the Printer, Number of Copies 5-44
Using SPOOK's HELP Facility 5-45

Chapter 6
Basic Account Structure

Chapter Overview 6-1
Account Structure Components 6-2
Accounts 6-2
Groups 6-2
Users 6-2
Files 6-2
Naming Conventions 6-4
Naming Rules 6-4
Fully Qualified File Names 6-4
The Group Name 6-4
The Directory 6-5
Functions of the Account Structure 6-5
Organization 6-6
Billing 6-6
Security 6-6
Passwords and Lockwords 6-7
Lockwords 6-10
File Access Permissions 6-11
Reassigning File Access Permissions 6-15
Capabilities 6-17

Chapter 7
Working with Files

Chapter Overview 7-1
Introduction 7-2
Summary 7-3
Basic Uses of the File System 7-3
Displaying Information about Files 7-4
Finding Files 7-4
Displaying a File's Contents 7-5
Renaming Files 7-6
Deleting (Purging) Files 7-7
Copying Files 7-8
Other Basic File Handling Tasks 7-8

Table of Contents (Continued)

Advanced Uses of the File System 7-9
File System Overview 7-9
Creating Files 7-9
File Characteristics 7-9
The :BUILD Command 7-10
Device Files 7-12
System-defined Files 7-12
The $STDIN and $STDLIST System-defined Files 7-13
The $NEWPASS, $OLDPASS, and $NULL System-defined Files 7-13
Subsystem Formal File Designators 7-14
Writing File Equations 7-15
Linking Actual Files with Generic File References 7-16
Creating Device Files 7-17
Backreferencing File Equations 7-18
Summary: Writing File Equations 7-19
Checking Existing File Equations 7-19
Clearing File Equations 7-20
Printing Files 7-20
Displaying File Characteristics 7-21
Displaying File Characteristics with the :LISTF Command 7-21
Displaying File Characteristics with the LISTDIR5 Utility 7-22
Displaying Information about Temporary Files 7-23
Changing Temporary Files to Permanent Files 7-24

Chapter 8
Working in Sessions

Chapter Overview 8-1
Introduction 8-2
Starting and Stopping Sessions 8-2
Logging On 8-2
Required Parameters of the :HELLO Command 8-3
Optional Parameters of the :HELLO Command 8-3
The Basic Logon Procedure 8-5
The Logon Message 8-6
Determining your Session Priority 8-7
Determining System Access (Input Priority) 8-7
Determining Output Priority 8-8
The :(COMMAND) LOGON Option 8-9
Logging Off 8-9
Correcting and Repeating Commands 8-10
Correcting Commands with the :REDO Command 8-11
Repeating Commands with the :REDO Command 8-12
Running Programs and Subsystems 8-13
Interrupting Commands and Programs 8-13
Unlocking the Keyboard 8-14
Interrupting a Subsystem 8-15
Temporarily Interrupting a Program 8-15
Aborting a Program 8-15
Monitoring the Session 8-16
Displaying the Current Date and Time 8-16
Displaying Availability of Special Commands 8-16
Displaying Information about the Current Session 8-17

Table of Contents (Continued)

Displaying Information about System Devices 8-18
Displaying Information about Jobs and Sessions 8-20
Sending and Receiving Messages 8-25
Messages from the Operator 8-25
The Welcome Message 8-25
The WARNING Message 8-25
Sending Messages 8-26
The :TELLOP Command 8-26
The :TELL Command 8-26
Turning Messages On and Off 8-27
Using the HELP Facility 8-27
Session Security 8-27

Chapter 9
Jobs

Overview 9-1
Introduction 9-2
The Batch Processing Commands 9-2
The Prompt 9-3
The !Job Command 9-3
The !COMMENT Command 9-7
The !TELL Command 9-7
The !TELLOP Command 9-7
The !CONTINUE Command 9-7
The !EOJ Command 9-8
Working with Jobs 9-8
Creating a Job File 9-8
Submitting (Streaming) the Job File 9-12
Monitoring a Job 9-13
Security of Batch Data 9-16
The Job Printout 9-17

Tables

Table 2-1. Summary of Recommended Terminal Settings 2-6
Table 2-2. Keys that Function Like Typewriter Keys 2-13
Table 2-3. "Special" Keys 2-14
Table 2-4. Summary of Command Sequences 2-17

Table 3-1. The Monitoring Commands 3-18
Table 3-2. The File Commands 3-19
Table 3-3. The Session Commands 3-20
Table 3-4. The Job Commands 3-21
Table 3-5. The Message Commands 3-21

Table 5-1. Subsystem Prompts and Functions 5-3
Table 5-2. Summary of EDITOR Commands 5-6
Table 5-3. Summary of Selected SPOOK Commands 5-40

Table 6-1. Default File Access Permissions 6-13
Table 6-2. User Codes for File Access Permissions 6-13
Table 6-3. Capabilities 6-18
Table 6-4. Default User Capabilities 6-19

Table 7-1. Subsystems and Files 7-2
Table 7-2. System Defined Files 7-13
Table 7-3. Sample Formal File Designators 7-15
Table 7-4. :LISTF Display Options 7-21

Table 8-1 The :REDO Subcommands 8-10

Table 9-1. Batch Processing Commands 9-3

Figures

Figure 1-1 The HP 150 and Vectra Microcomputers 1-2
Figure 1-2. Series of the HP 3000 1-3
Figure 1-3. Computer and Peripherals 1-4
Figure 1-4. Disc Drives 1-6
Figure 1-5. Tape Drives 1-7
Figure 1-6. System Printers 1-8
Figure 1-7. Sessions and Jobs:The Concept 1-10

Figure 2-1. Power Cable Connection for the HP 2392A Terminal 2-3
Figure 2-2. Direct Computer Connection for HP 2392A Terminal 2-4
Figure 2-3. Modem Computer Connection for HP 2392A Terminal 2-4
Figure 2-4. The Keyboard Cable Connection for the HP 2392A Terminal 2-5
Figure 2-5. Default Terminal Settings 2-8
Figure 2-6. Checking Terminal Modes Settings 2-9
Figure 2-7. Checking Configuration Settings (1) 2-10
Figure 2-8. Checking Configuration Settings (2) 2-10
Figure 2-9. Scrolling Data into the Screen Buffer 2-12
Figure 2-10. HP 2392A Standard USASCII Keyboard 2-13
Figure 2-11. The Function Key Label Screen 2-19
Figure 2-12. Programming the Function Keys 2-20
Figure 2-13. Using the Programmed Function Keys 2-21
Figure 2-14. The Printer Cable Connection for the HP 3292A Terminal 2-22
Figure 2-15. Printing a Screen Dump (1) 2-22
Figure 2-16. Printing a Screen Dump (2) 2-23

Figure 3-1. Anatomy of a Command 3-3
Figure 3-2. Parameters 3-4
Figure 3-3. Relationship between Positional, Keyword, and Subparameters 3-6
Figure 3-4. Delimiters 3-7
Figure 3-5. The Ampersand Continuation Character 3-8
Figure 3-6. :HELLO Command Syntax 3-9
Figure 3-7. :PURGE Listing in MPE Quick Reference Guide 3-11

Figure 4-1. Anatomy of a UDC 4-3
Figure 4-2. UDC File: The Concept 4-6

Figure 5-1. Anatomy of a Scratch File 5-5
Figure 5-2. Using the /GATHER Command 5-13
Figure 5-3. Using the /COPY Command (1) 5-14
Figure 5-4. Using the /COPY Command (2) 5-15
Figure 5-5. Employee File with Records and Fields 5-21
Figure 5-6. The FCOPY Subsystem 5-34

Figure 6-1. Account Component Relationships 6-3
Figure 6-2. The Fully Qualified File Name 6-4
Figure 6-3. Basic Logon Sequence 6-5
Figure 6-4. Logon Sequence with Optional Group 6-5
Figure 6-5. Passwords and Lockwords 6-8
Figure 6-6. The :HELLO Command with Passwords 6-9
Figure 6-7. File Access Permissions at the System, Group, and File Levels 6-12
Figure 6-8. Restricting File Access Permissions 6-16

Figures (Continued)

Figure 7-1. Overview of the File System 7-9
Figure 7-2. Syntax of the :BUILD Command 7-11
Figure 7-3. The $NEWPASS and $OLDPASS System-defined Files 7-14
Figure 7-4. The :FILE Command 7-15

Figure 8-1. The :HELLO Command 8-3
Figure 8-2. The Logon Passwords 8-4
Figure 8-3. The Optional Session Name 8-4

Figure 9-1. The !JOB Command 9-4
Figure 9-2. The Passwords 9-5
Figure 9-3. The Job Name 9-5
Figure 9-4. Anatomy of a Job File 9-9
Figure 9-5. The Trailer Page on the Job Printout 9-18

Examples

Example 2-1. Printing a Listing on the Line Printer 2-24
Example 3-1. Step 1:Displaying General Information 3-13
Example 3-2. Steps 2, 3, and 4:Displaying General Information, Displaying a Command, and Exiting the HELP Facility 3-14
Example 3-3. Displaying Command Syntax 3-15
Example 3-4. Displaying Information about Parameters 3-16
Example 3-5. Displaying Information about how a Command Works 3-16
Example 3-6. Displaying a Command Example 3-17
Example 4-1. Simple UDC Example 4-3
Example 4-2. Creating several UDCs within a UDC File 4-8
Example 4-3. Nesting UDCs 4-10
Example 4-4. Modifying a UDC 4-13
Example 4-5. Deleting a UDC from the UDC File 4-15
Example 4-6. Displaying UDCs with the :SHOWCATALOG Command 4-17
Example 4-7. Using the HELP Facility to Display UDC Contents 4-18
Example 5-1. Creating a New File with EDITOR 5-8
Example 5-2. Modifying an Existing File with EDITOR 5-10
Example 5-3. Inserting Text and Renumbering Lines in an Existing File 5-12
Example 5-4.. Moving Lines Within a File 5-14
Example 5-5. Copying Lines Within a File 5-16
Example 5-6. Deleting Lines Within a File 5-18
Example 5-7. Printing a File 5-19
Example 5-8. Using EDITOR's HELP Facility 5-20
Example 5-9. Creating a File to be Sorted 5-23
Example 5-10. Sorting a File and Displaying the Sorted File 5-25
Example 5-11. Merging and Displaying Two Sorted Files 5-27
Example 5-12. Copying a Single File to Tape with the :STORE Command 5-29
Example 5-13. Copying Several Files to Tape with the :STORE Command 5-30
Example 5-14. Using the SHOW Parameter in the :STORE Command 5-31
Example 5-15. Using the PURGE Parameter in the :STORE Command 5-32
Example 5-16. Using the PROGRESS Parameter in the :STORE Command 5-32
Example 5-17. Using the :RESTORE Command 5-33
Example 5-18. Creating a New File with the :FCOPY Command 5-36
Example 5-19. Using FCOPY to Copy between Accounts on the Same System 5-37
Example 5-20. Using the :FCOPY Command to Compare File Contents 5-38
Example 5-21. Using the FCOPY Command to Print a File 5-39
Example 5-22. Sample SHOW Screen 5-41
Example 5-23. Displaying a Spoolfile 5-42
Example 5-24. Locating Data in Spoolfiles 5-43
Example 5-25. Adding Data to Spoolfiles 5-44
Example 5-26. Deleting Spoolfiles 5-44
Example 5-27. Changing a Spoolfile's Output and Number of Copies Printed 5-45
Example 5-28. Using the SPOOK HELP Facility 5-45
Example 6-1. Changing User Passwords 6-10
Example 6-2. Adding Lockwords to Existing Files 6-11
Example 6-3. Changing Lockwords on Existing Files 6-11
Example 6-4. Displaying File Access Permissions 6-15
Example 6-5. Displaying User Capabilities 6-20
Example 7-1. Sample :LISTF Display 7-4
Example 7-2. Using Wild Card Characters to Find Files 7-5
Example 7-3. Renaming a File 7-7

Examples (Continued)

Example 7-4. Deleting (Purging) a File 7-8
Example 7-5. Creating a File with the :BUILD Command 7-12
Example 7-6. Writing File Equations to Link Formal File Designators with Actual File Designators 7-17
Example 7-7. Creating Device Files to Reroute Output Files 7-17
Example 7-8. Backreferencing to a Device File 7-18
Example 7-9. Comparison to Example 7-8 7-18
Example 7-10. Displaying Existing File Equations 7-19
Example 7-11. Deleting File Equations 7-20
Example 7-12. :LISTF Display with Parameter 1 7-22
Example 7-13. Using the LISTDIR5 Utility 7-23
Example 7-14. Displaying Temporary Files 7-24
Example 7-15. Saving a Temporary File 7-24

Example 8-1 Sample Logon Procedure and Logon Message 8-7
Example 8-2. Changing the Input Priority 8-8
Example 8-3. Using the :(Command) Logon Option 8-9
Example 8-4. Sample Logoff Procedure and Logoff Message 8-10
Example 8-5. Using the :REDO Command to Correct Errors 8-12
Example 8-6. Running Programs and Subsystems 8-13
Example 8-7. Sample :SHOWTIME Display 8-16
Example 8-8. Sample :SHOWALLOW Display 8-17
Example 8-9. Sample :SHOWMEDisplay 8-18
Example 8-10. Sample Partial :SHOWDEV Display 8-19
Example 8-11. Sample :SHOWDEV Display, with Parameter 8-19
Example 8-12. Sample :SHOWJOB SCHED Display 8-20
Example 8-13. Sample :SHOWJOB Display, without Parameters 8-21
Example 8-14. Sample :SHOWOUT STATUS Display 8-23
Example 8-15. Sample :SHOWOUT Display 8-23
Example 8-16. Sample WARNING Message from Operator 8-25
Example 8-17. Sample Message to Operator 8-26
Example 8-18. Sample Message to Other User 8-26

Example 9-1. Sample !JOB Command, with Parameters 9-7
Example 9-2. Creating a Job File with EDITOR 9-11
Example 9-3. Streaming a Job File 9-12
Example 9-4. Streaming a Job from the Terminal 9-13
Example 9-5. Sample :SHOWJOB Display, without Parameters 9-14
Example 9-6. :SHOWJOB Command, with Job Number Parameter 9-16
Example 9-7. Securing a Job File with the :ALTSEC Command 9-17

Procedures

Chapter 2
The Terminal

To connect the terminal to a power source: 2-3
To check the computer cable connection: 2-4
To check the keyboard cable connection: 2-5
To check the terminal mode settings: 2-8
To check the configuration settings: 2-9
To display text scrolled off the top of the screen: 2-12
To display text scrolled off the bottom of the screen: 2-12
To control the terminal with a control sequence: 2-16
To control the terminal with an escape sequence: 2-16
To program the function keys: 2-19
To use the programmed function keys: 2-20
To connect the printer cable to the terminal: 2-21
To print a screen dump on your workstation printer: 2-22
To print a listing: 2-23

Chapter 3
Introduction to Commands

To print a screen dump on your workstation printer: 2-22
To find a command with the HELP facility: 3-13
To display a command's syntax with the HELP facility: 3-15
To display information about a command's parameters with the HELP facility: 3-15
To display information about a command's operation with the HELP facility: 3-16
To display an example of the command with the HELP facility: 3-17

Chapter 4
UDCs

To create a UDC file: 4-7
To create nested UDCs: 4-9
To add a UDC to a UDC file: 4-11
To modify a UDC in a UDC file: 4-12
To delete a UDC from a UDC file: 4-14
To display a list of all UDC files and the UDCs they contain: 4-16
To find out what each UDC does: 4-17

Chapter 5
Utilities and Subsystems

To create a new file: 5-7
To modify an existing file: 5-9
To insert text between lines: 5-11
To move lines within text: 5-13
To copy lines within the file: 5-15
To delete lines from a file: 5-17
To print a file: 5-19
To use EDITOR's HELP facility: 5-20
To create a file to be sorted: 5-22
To sort files: 5-24

Procedures (Continued)

To merge two sorted files: 5-26
To "store" a file to tape: 5-29
To restore a file from tape: 5-33
To copy files between groups or accounts with the FCOPY subsystem: 5-36
To compare the contents of two files: 5-38
To print a file with FCOPY: 5-38
To display spoolfiles: 5-41
To find a character string: 5-43
To append two files: 5-43
To delete a spoolfile: 5-44
To change the printer: 5-44
To change the number of copies printed: 5-44
To use the SPOOK HELP Facility: 5-45

Chapter 6
Basic Account Structure

To create or change a password 6-9
To add a lockword to an existing file 6-11
To change a lockword: 6-11
To check file access permissions: 6-14
To restrict file access permissions: 6-16
To temporarily remove restrictions to file access permissions: 6-17
To reinstate restrictions to file access permissions: 6-17
To display user capabilities 6-19

Chapter 7
Working with Files

To rename a file: 7-6
To rename a file and add a lockword: 7-6
To delete a file from disc: 7-7
To write a file equation to link formal and actual file designators: 7-16
To create a device file with a file equation: 7-17
To backreference a file equation: 7-18
To check file equations set to the current session: 7-19
To clear a file equation: 7-20
To clear all file equations: 7-20
To display a file's code, record size, end-of-file location, and maximum number of records: 7-21
To display a file's characteristics: 7-22
To list temporary files: 7-23
To "save" a user defined temporary file: 7-24
To "save" the system-defined file $OLDPASS: 7-24

Chapter 8
Working with Sessions

To log on: 8-6
To change your input priority: 8-8
To use the :(COMMAND) LOGON option: 8-9
To log off: 8-10
To delete selected characters: 8-11
To insert characters: 8-11

Procedures (Continued)

To replace characters: 8-11
To cancel the most recent change: 8-11
To cancel all changes made since the command was last executed: 8-11
To repeat commands: 8-12
To unlock the keyboard with a soft reset: 8-14
To unlock the keyboard with a hard reset: 8-14
To interrupt a subsystem: 8-15
To temporarily interrupt a program: 8-15
To abort a program: 8-15
To display the current date and time: 8-16
To find out what special commands are available to you: 8-17
To display information about the current session: 8-18
To display the status of all devices: 8-19
To display the status of a particular device: 8-19
To display information about scheduled jobs only: 8-20
To display information about all jobs and sessions: 8-20
To see a summary of spoolfiles: 8-23
To see what spoolfiles exist for your session: 8-23
To send a message to the Operator: 8-26
To send a message to other users: 8-26
To block messages: 8-27
To receive messages again: 8-27

Chapter 9
Jobs

To create a job file: 9-10
To submit a job file for processing: 9-12
To stream a job from the terminal: 9-12
To check on a job's status: 9-15
To prevent others from looking at a job file: 9-17

Conventions

NOTATION	DESCRIPTION
UPPERCASE	Within syntax statements, characters in uppercase must be entered in exactly the order shown, though you can enter them in either uppercase or lowercase. For example: `SHOWJOB` Valid entries: `showjob` `ShowJob` `SHOWJOB` Invalid entries: `shojwob` `Shojob` `SHOW_JOB`
italics	Within syntax statements, a word in italics represents a formal parameter or argument that you must replace with an actual value. In the following example, you must replace *filename* with the name of the file you want to release: `RELEASE` *filename*
punctuation	Within syntax statements, punctuation characters (other than brackets, braces, vertical parallel lines, and ellipses) must be entered exactly as shown.

{ }	Within syntax statements, braces enclose required elements. When several elements within braces are stacked, you must select one. In the following example, you must select `ON` or `OFF`:
	
	`SETMSG {ON }`
	`{OFF}`

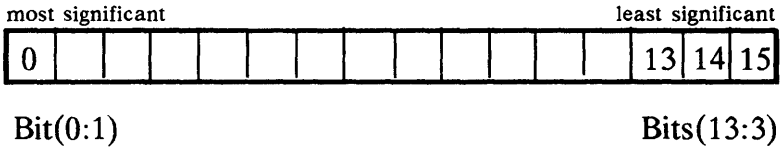
[ ]	Within syntax statements, brackets enclose optional elements. In the following example, brackets around `,TEMP` indicate that the parameter and its delimiter are optional: `PURGE` *filename*[`,TEMP`] When several elements within brackets are stacked, you can select any one of the elements or none. In the following example, you can select *devicename* or *deviceclass* or neither: `SHOWDEV` [*devicename*] [*deviceclass*]

Conventions (Continued)

NOTATION	DESCRIPTION
[...]	Within syntax statements, a horizontal ellipsis enclosed in brackets indicates that you can repeatedly select elements that appear within the immediately preceding pair of brackets or braces. In the following example, you can select *itemname* and its delimiter zero or more times. Each instance of *itemname* must be preceded by a comma: [,*itemname*][...] If a punctuation character precedes the ellipsis, you must use that character as a delimiter to separate repeated elements. However, if you select only one element, the delimiter is not required. In the following example, the comma cannot precede the first instance of *itemname*: [*itemname*][,...]
|...|	Within syntax statements, a horizontal ellipsis enclosed in parallel vertical lines indicates that you can select more than one element that appears within the immediately preceding pair of brackets or braces. However, each element can be selected only one time. In the following example, you must select `,A` or `,B` or `,A,B` or `,B,A` : {`,A`} {`,B`} |...| If a punctuation character precedes the ellipsis, you must use that character as a delimiter to separate repeated elements. However, if you select only one element, the delimiter is not required. In the following example, you must select `A` or `B` or `A,B` or `B,A` . The first element cannot be preceded by a comma: {`A`} {`B`}|,...|
... ⋮	Within examples, horizontal or vertical ellipses indicate where portions of the example are omitted.
△	Within syntax statements, the space symbol △ shows a required blank. In the following example, you must separate *modifier* and *variable* with a blank: `SET`[(*modifier*)]△(*variable*);

Conventions (Continued)

NOTATION	DESCRIPTION
[ ]	The symbol [ ] indicates a key on the terminal's keyboard. For example, [CTRL] indicates the Control key.
[CTRL] *char*	[CTRL] *char* indicates a control character. For example, [CTRL] Y means you have to simultaneously press the Control key and the Y key on the keyboard.
base prefixes	The prefixes %, #, and $ specify the numerical base of the value that follows: *%num* specifies an octal number *#num* specifies a decimal number $*num* specifies a hexadecimal number When no base is specified, decimal is assumed.
Bit (*bit:length*)	When a parameter contains more than one piece of data within its bit field, the different data fields are described in the format Bit (*bit:length*), where *bit* is the first bit in the field and *length* is the number of consecutive bits in the field. For example, Bits (13:3) indicates bits 13, 14, and 15:



1

Basic Concepts

Chapter Overview

This chapter introduces you to the HP 3000 Computer System. It gives an overview of the different parts that make up the system and explains some basic concepts. It includes the following topics:

- Types of computers
- HP 3000 hardware components
- HP 3000 software components
- Introduction to working with the HP 3000

Types of Computers

There are many different types of computers, which are generally classified as microcomputers, minicomputers, and mainframe computers.

Microcomputers

Microcomputers are also referred to as personal computers or PCs. These PCs are generally used by single users at their desks or workstations. A wide variety of programs is available for wordprocessing, spreadsheets, games, and many other uses.

For loading and storing programs and data, PCs use floppy discs inserted into PC disc drives, or hard discs built into the PC. Most PCs are also attached to a printer that lets users print copies of their files at their desks.

PCs can also be used as "terminals" that communicate with minicomputers or mainframe computers, as explained in more detail in Chapter 2, "The Terminal." In this case, you do not use the PCs internal memory or processing power. Instead, you simply use the PC keyboard and screen to interact with the HP 3000 minicomputer.

Hewlett-Packard offers several PC models, including the HP 150 and the Vectra, shown in Figure 1-1. Both models may be ordered with either floppy disc drives or internal hard discs.



LG200077_001

Figure 1-1. The HP 150 and Vectra Microcomputers

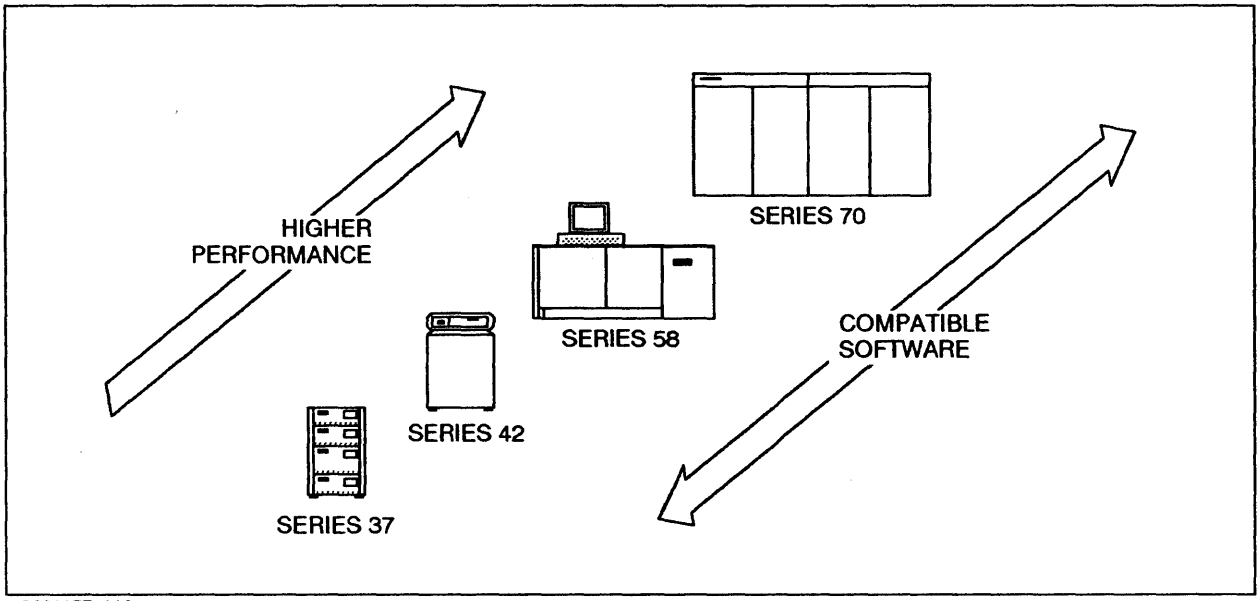
Minicomputers

The HP 3000 is considered a minicomputer. Minicomputers have much more internal memory and processing power than PCs. This makes it possible for many users to share the computer at the same time, for many different tasks. For that reason, the HP 3000 is called a "multiuser system." You may be able to tell that the computer responds more slowly during "busy" times when many different people are trying to use it.

Unlike PCs, minicomputers can also handle several tasks at the same time, which is known as "multiprocessing." Because so many people use the system, floppy discs or internal disc drives are not sufficient to store all the programs and data users need. For that reason, separate disc drives are attached to the computer for storage.

Hewlett-Packard offers several minicomputers. The most popular business system is the HP 3000. There are several series of the HP 3000, as shown in Figure 1-2.

These systems work in the same way, but they differ in the number of users they can handle, the amount of information they can process, the processing speed, and so on.



LG200077-002

Figure 1-2. Series of the HP 3000

Mainframe Computers

Mainframe computers have even more power than minicomputers, although it may be difficult to draw the line between mainframe computers and some of today's powerful minicomputers. In general, mainframe computers are used to run large batch jobs (such as generating monthly utility bills), maintaining central databases (such as an airline reservations system), or controlling and coordinating several other computers.

Hardware Components of a Computer System

Any computer system consists of hardware and software components. The hardware is the physical equipment; that part of the system that looks like boxes of various shapes and sizes. The software consists of the programs that make the system run.

The actual computer is just one part of the system. You also need other equipment to communicate with the computer. This equipment is called "peripherals," and individual peripherals are referred to as "devices." Figure 1-3 shows a typical computer system.

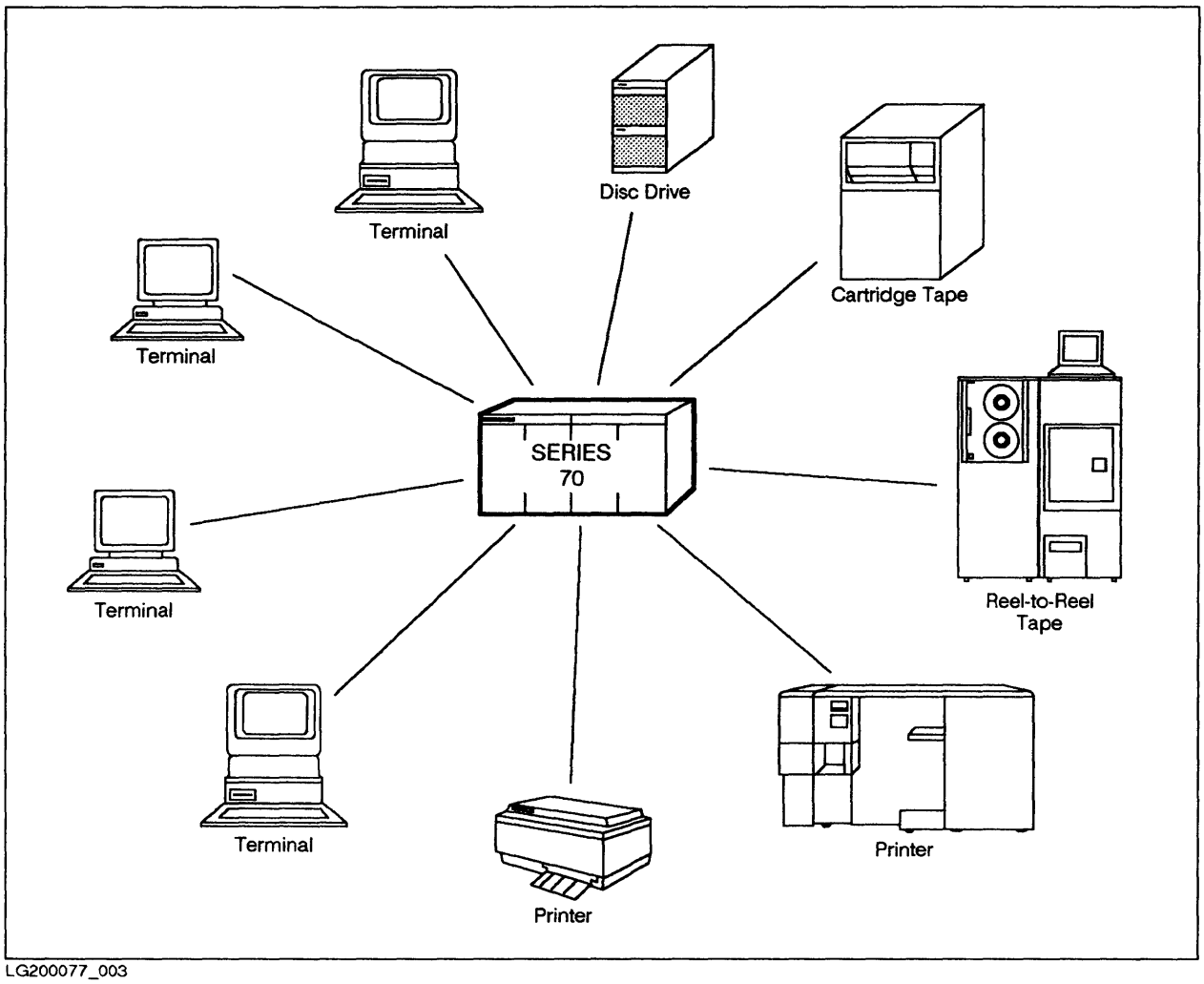


Figure 1-3. Computer and Peripherals.

The HP 3000 Computer

This is the "heart" of the system. It is often referred to as the CPU (Central Processing Unit), which is the general name for that part of the computer that contains the circuits where the actual processing takes place. For example, when you tell the computer to add a list of numbers, it happens in the CPU.

Depending on which series of the HP 3000 you have, the box that contains the CPU may look like any of systems shown in Figure 1-2.

Peripherals

A system's peripherals usually include one or more of the following devices: terminals, disc drives, tape drives, and printers. Each device in an HP 3000 system is assigned a number that identifies it to the system. This number is called the "ldev" number.

Terminals

Terminals are the devices that let you communicate with the CPU. They consist of a screen and an attached keyboard. You type information to be sent to the computer on the keyboard. This information is displayed on the screen as you type. Information from the computer is also displayed on the screen. Chapter 2, "The Terminal" describes the terminal in more detail.

In addition to the terminals for individual users, one terminal is used by the System Operator to control the entire system. This terminal is called the "System Console." It is usually located near the CPU.

Disc Drives

Information not being processed at the moment is stored on disc drives. You can store information on disc and then retrieve it very quickly, in much the same way that music is recorded on a record that can then be played. Depending on the needs of the system, one or more disc drives may be attached to the CPU.

Disc drives are inside cabinets that look similar to Figure 1-4.

Peripherals

A system's peripherals usually include one or more of the following devices: terminals, disc drives, tape drives, and printers. Each device in an HP 3000 system is assigned a number that identifies it to the system. This number is called the "ldev" number.

Terminals

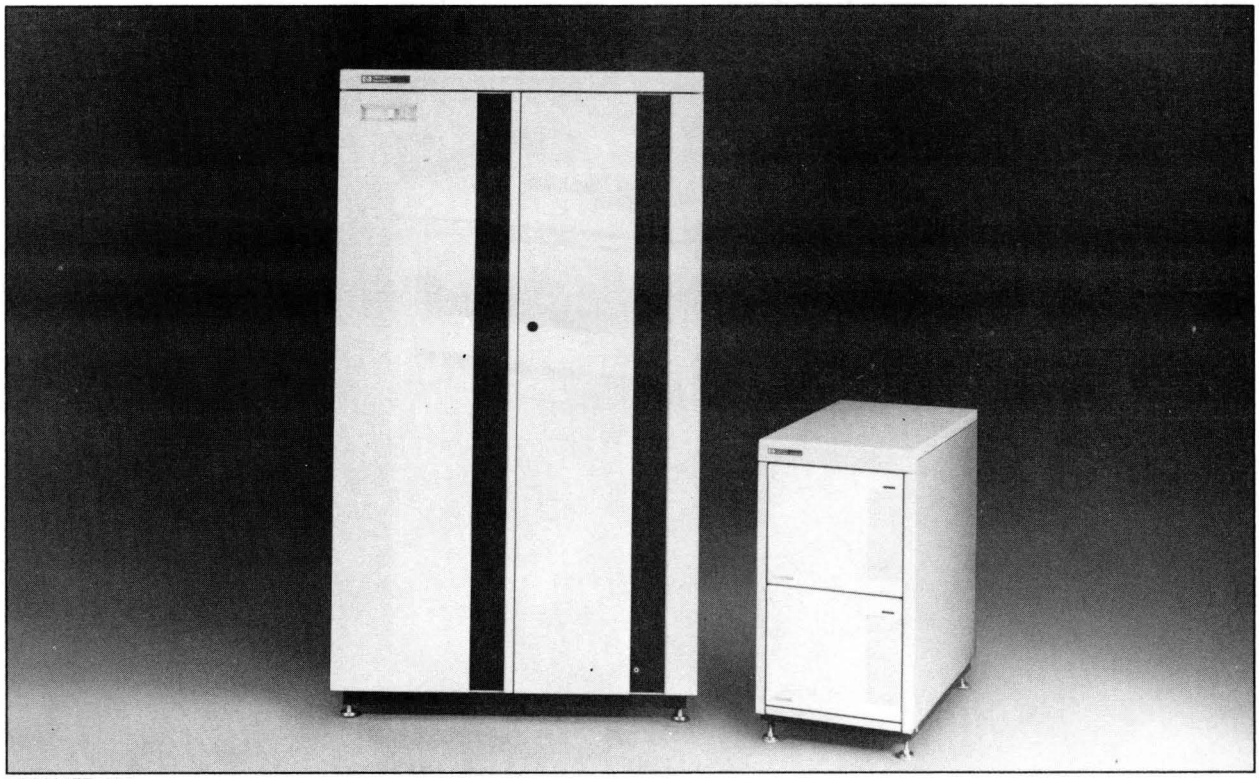
Terminals are the devices that let you communicate with the CPU. They consist of a screen and an attached keyboard. You type information to be sent to the computer on the keyboard. This information is displayed on the screen as you type. Information from the computer is also displayed on the screen. Chapter 2, "The Terminal" describes the terminal in more detail.

In addition to the terminals for individual users, one terminal is used by the System Operator to control the entire system. This terminal is called the "System Console." It is usually located near the CPU.

Disc Drives

Information not being processed at the moment is stored on disc drives. You can store information on disc and then retrieve it very quickly, in much the same way that music is recorded on a record that can then be played. Depending on the needs of the system, one or more disc drives may be attached to the CPU.

Disc drives are inside cabinets that look similar to Figure 1-4.



LG200077_004

Figure 1-4. Disc Drives

Tape Drives

Information may also be stored on tapes. However, since the computer has to search for the information starting at the beginning of the tape, it takes much longer to retrieve information from tape than from disc. For that reason, tape is generally used only to "back up" information stored on disc. That way, if something happens to that information, the operations staff can restore any data that may have been lost. Also, information stored on tapes is easy to transport.

Tape drives transfer information between discs and tapes. There are two basic kinds of tape drives: cartridge tape drives, which use cartridge tapes, and reel-to-reel tape drives, which use reel tapes. Figure 1-5 shows a tape drive.



LG200077_005

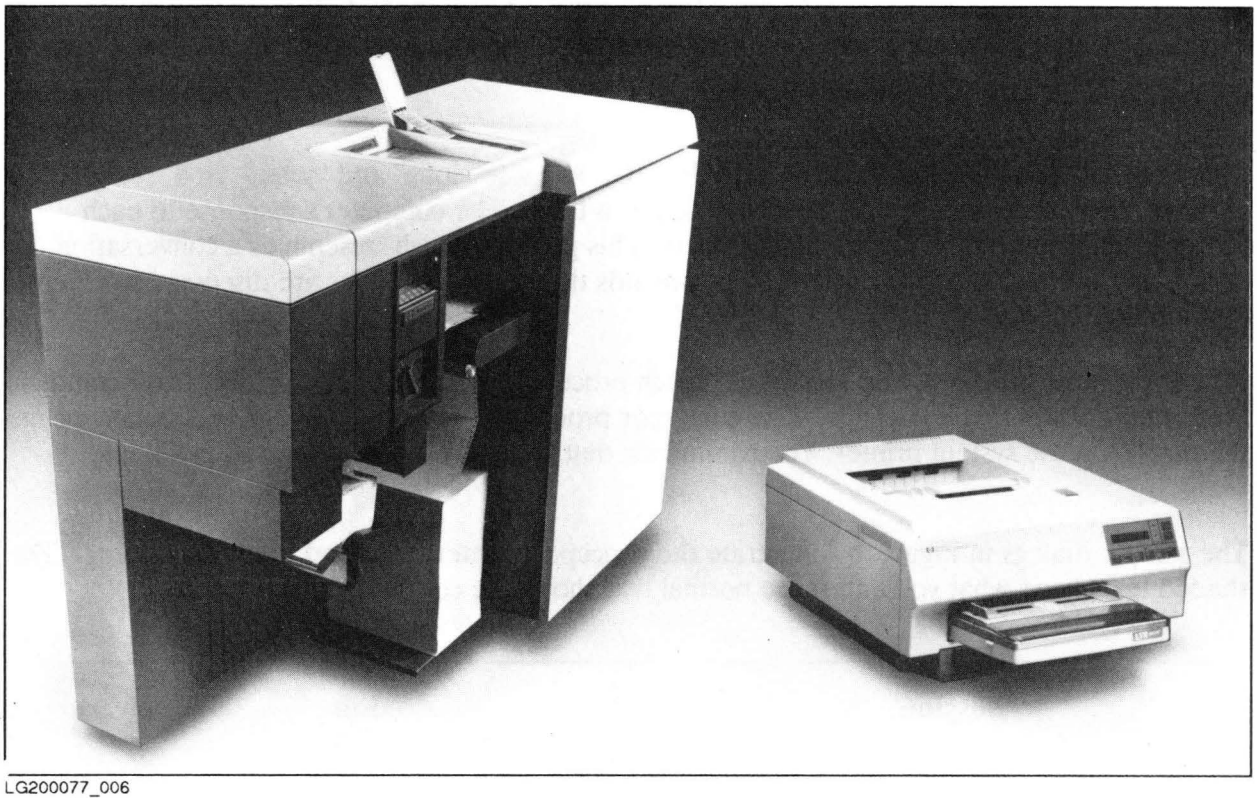
Figure 1-5. Tape Drives

Printers

Printers come in all shapes and sizes. They also vary in speed, as well as the quality of the printouts they produce. For example, Hewlett-Packard's Laser Printer (Model 2680) printouts are near typeset quality.

Depending on the number of users on the system, one or more printers may be attached to the CPU. As users request printouts, the system creates a waiting list, which it satisfies in the order received.

Printers look similar to Figure 1-6.



LG200077_006

Figure 1-6. System Printers

Software Components of a Computer System

The software components are the programs that make the computer run and determine whether the computer acts as a graphic artist, accountant, text processor, office manager, or any of hundreds of applications. These programs translate the user's instructions to the language the computer understands, which consists of huge numbers of 0s and 1s (called "binary numbers").

The HP 3000 is shipped with software called the "FOS," which stands for "Fundamental Operating Software". The FOS contains the software necessary for most processing tasks. It includes:

- The MPE (Multi-Programming Executive) operating system.
- An information management system (IMAGE).
- Several subsystems and utilities to perform specific tasks.

You can also purchase additional software to meet specific needs, including special programs (called "compilers") that let you write your own software. Each programming language has its own compiler.

Overview: Working with the HP 3000

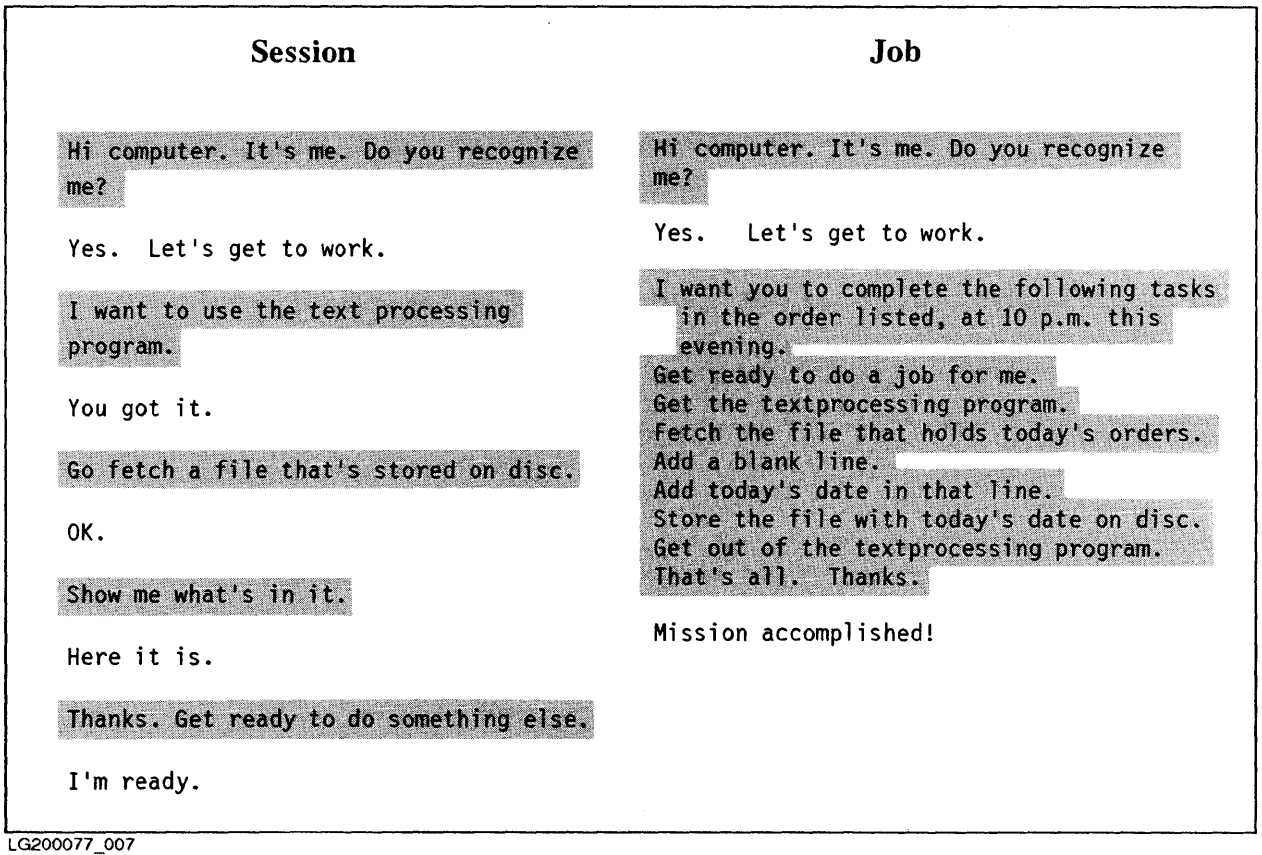
This manual describes how to work with MPE and the subsystems included with the FOS.

Sessions and Jobs

There are two basic ways of working with the HP 3000: "sessions" and "jobs." In a session, you enter commands at the terminal keyboard one at a time. The computer's response to each command is displayed on the terminal screen. This process, which resembles a conversation, is known as "interactive processing." The commands that control sessions are discussed in Chapter 8, "Working in Sessions."

When you work with jobs, also known as "batch processing," you submit a series of commands to the computer as a single package. The computer processes the entire package and usually prints the results on the system printer. The commands that control jobs are discussed in Chapter 9, "Jobs."

The sample dialogs in Figure 1-7 illustrate the concepts of interactive and batch processing. The shaded text shows what you enter; the normal text shows the computer's response.



LG200077_007

Figure 1-7. Sessions and Jobs: The Concept

Defaults

Whether you work in a session or a job, you can decide over how much control you have over the system. When the HP 3000 is shipped from the factory, the MPE operating system, the FOS, and any peripheral devices are already preset to perform in the most commonly used way. These preset operating characteristics are called "defaults." Defaults make it easier to use the system because you can leave many decisions up to the computer.

For example, the default for saving a file is to store it on disc, since that is what most users want most of the time. Therefore, all files are automatically stored on disc unless you specify otherwise.

This manual describes how to accomplish many basic tasks using the system defaults. It also introduces many of the available options that give you more precise control over the system, to meet specific needs. There are times, for example, when you want to store a file on tape or print it on a printer instead of storing it on disc.

2

The Terminal

Chapter Overview

This chapter introduces you to the terminal, using the popular HP 2392A terminal as an example. However, the concepts and procedures in this chapter also apply, with minor differences, to other widely used terminals, such as the HP 700/92 terminal and the HP 150 and Vectra microcomputers when used as terminals.

This chapter includes the following topics:

- Basic concepts
- Things to check, including connections and terminal settings
- The concept of scrolling
- A comparison of a typewriter and the terminal keyboard
- The "special" keys
- How to program the function keys
- How to print what is on the screen

Introduction

The terminal is your connection with the HP 3000. You can use many different terminals, including PCs that can also be used as computers in their own right. Basically, all terminals work the same, although the details may vary from model to model. For example, the key that sends information to the computer is marked "RETURN" on some terminals, "ENTER" on others.

The terminal has two main parts; the keyboard and the screen. The terminal's keyboard is your means of "talking" to the HP 3000. You use it to enter commands, as well as data to be processed. Since the keyboard lets you enter information to be sent to the computer, a terminal is considered an "input device."

The terminal's screen normally displays any information you enter on the keyboard. The screen can also display information stored on the discs connected to the CPU, as well as messages from the computer, the Operator, or other users. Since the screen displays information from the computer, the terminal is considered an "output device" as well as an input device.

When you work with the computer, it will seem as if things are happening on your terminal screen. However, it is important to realize that information is being processed wherever the HP 3000's CPU is physically located. This is true whether the terminal has its own CPU (in the case of a PC) or whether it is a "dumb" terminal that cannot do much on its own.

Things to Check

Chances are that the person who set up your terminal already checked the connections and made sure that the terminal settings were correct. However, if you have to set up your terminal yourself, you will need the information that follows. Also, connections can become loose over time, and other people can change terminal settings to suit their own needs. If your usual procedures do not work as expected, you may want to check these items before calling for help.

NOTE

Terminal settings are maintained by a battery with a life span of approximately two years. If you find that your settings have changed inexplicably, have someone check this battery.

Connections

Your terminal must be connected to a power source and to the HP 3000. In addition, the keyboard needs to be connected to the main body of the terminal. If you have problems locating these connections, or if you need more information about the cables, refer to the documentation that came with the terminal or ask your manager.

CAUTION

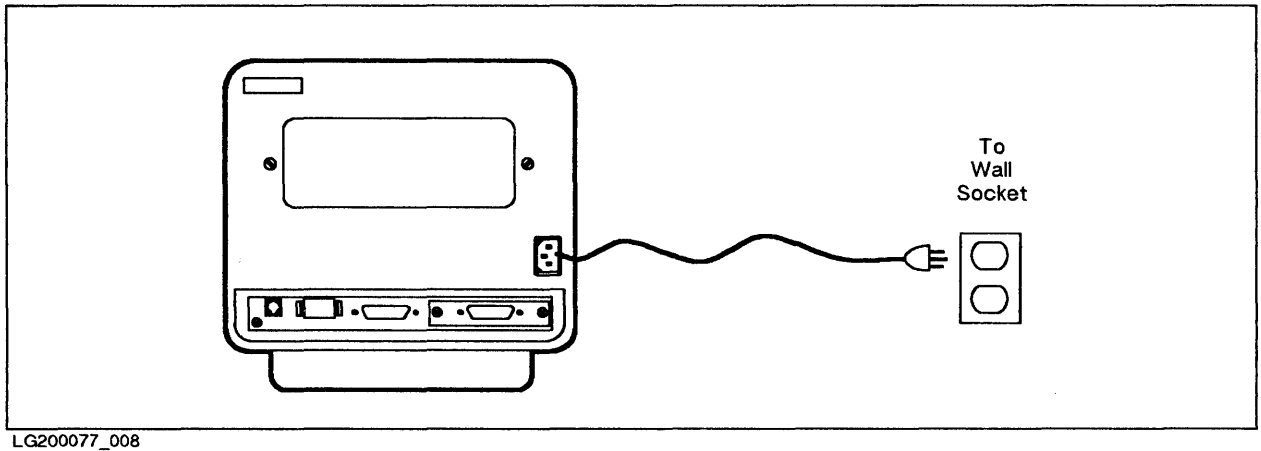
Make sure the terminal is turned off before connecting the keyboard to the main body of the terminal or before connecting any cables to the back of the terminal.

Power Cable Connection

When you turn on the terminal, you should hear the fan immediately and see the cursor flashing after a minute or so. If you get no response, check the power cable connection. The power cord may become detached from either the back of the terminal or from the wall outlet.

To connect the terminal to a power source:

1. Turn off the terminal.
2. Plug one end of the power cable into the terminal. Refer to Figure 2-1 to locate the power cable connector on the HP 2392A terminal. If you have a different terminal, refer to its documentation.
3. Insert the plug into a wall outlet.



LG200077_008

Figure 2-1. Power Cable Connection for the HP 2392A Terminal

Computer Cable Connection

The terminal must also be connected to the computer with a computer cable. This cable may connect the terminal and the HP 3000 directly, as shown in Figure 2-2. However, many systems use a "modem" connection, which means that the terminal actually communicates with the computer over telephone wires. In this case, the computer cable attaches to the modem, which in

turn is connected to a telephone jack or outlet. It may also be directly connected to the phone (PBX). A typical modem connection is shown in Figure 2-3.

To check the computer cable connection:

1. Turn off the terminal.
2. Make sure the computer cable is firmly attached to the terminal. If necessary, tighten the screws. Refer to Figure 2-2 to locate the computer cable connection on the HP 2392A terminal. If you have a different terminal, refer to its documentation or ask your manager.

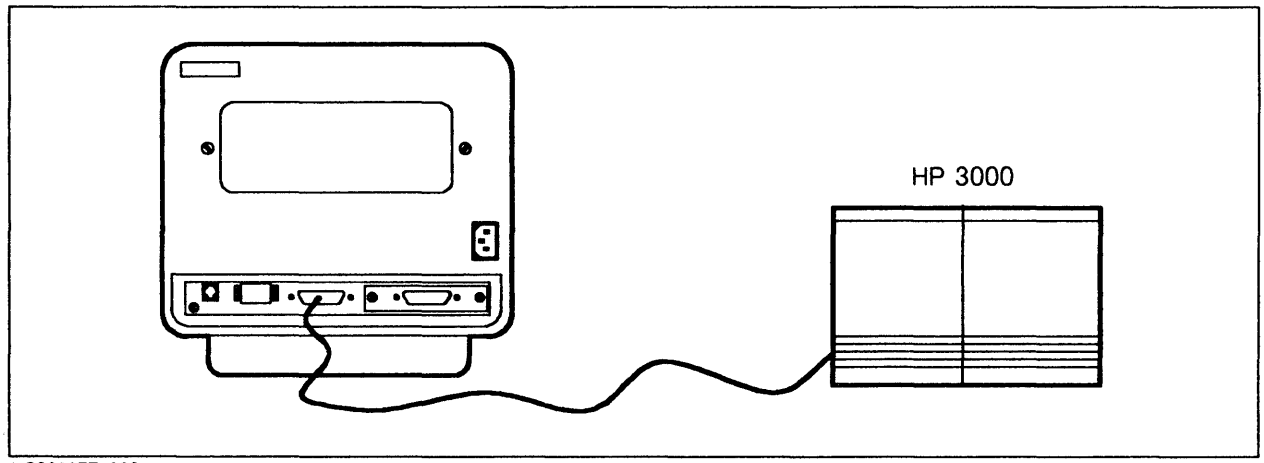


Figure 2-2. Direct Computer Connection for HP 2392A Terminal

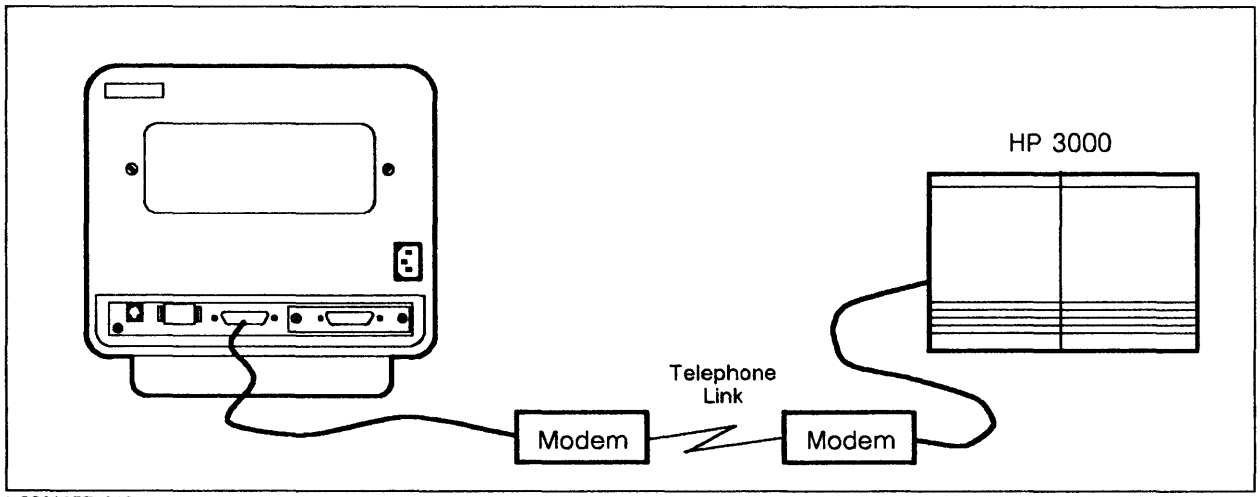


Figure 2-3. Modem Computer Connection for HP 2392A Terminal

NOTE

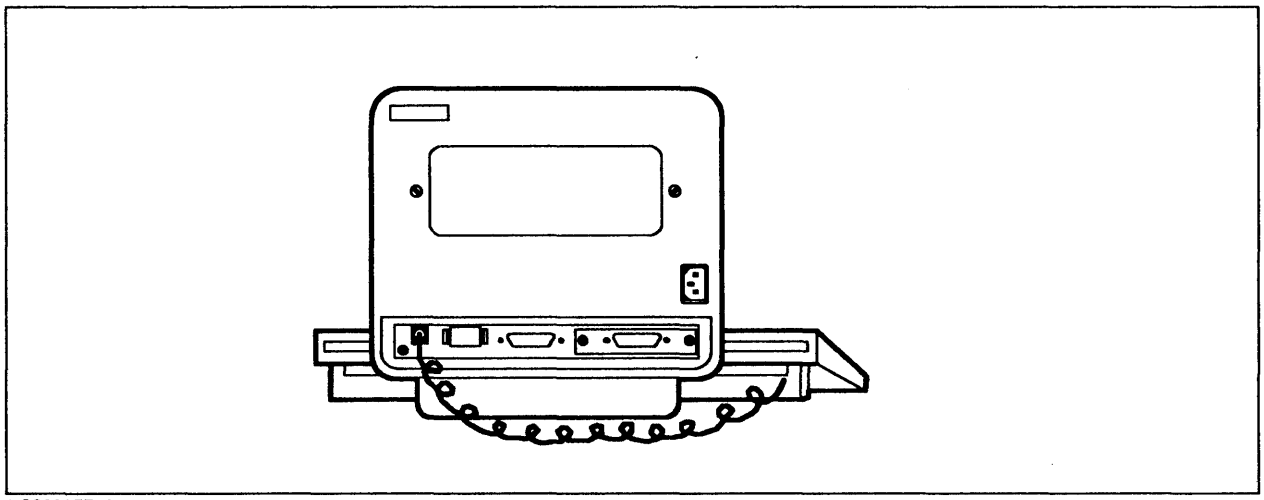
In PCs, the modem may be built in. If you are using a PC with a built-in modem, the PC will be connected to the telephone.

Keyboard Cable Connection

If nothing appears on the screen when you type, first check to make sure the keyboard cable is connected to the terminal.

To check the keyboard cable connection:

1. Turn off the terminal.
2. Plug the modular, jack-like connector at the end of the keyboard cable into the back of the terminal Refer to Figure 2-4 to locate the keyboard cable connection on the HP 2392A terminal. Note that other keyboards may have a jack-like connector on the keyboard as well.



LG200077_011

Figure 2-4. The Keyboard Cable Connection for the HP 2392A Terminal

Terminal Settings

In addition to the power, computer, and keyboard cable connections, certain terminal settings have to be set to specific values. Terminal settings are already preset at the Hewlett-Packard factory to the values needed by most users (defaults). If your organization requires customization, refer to the terminal's documentation for complete information.

This chapter only covers those settings that *must* be set a certain way to allow the terminal to communicate with the HP 3000. This includes the Remote mode, Block mode, and Auto LF settings, the Baud Rate and Parity settings, and the Enq/Ack setting. Once set, these settings remain until someone changes them, even when the terminal is turned on and off.

Table 2-1 summarizes the terminal settings and their correct values. The next section briefly explains what the settings mean. However, you don't have to know what they mean to be able to check or change them. You may want to read the short explanations that follow or skip to the instructions for checking and changing the terminal settings.

Table 2-1. Summary of Recommended Terminal Settings

Setting	Value
REMOTE MODE	ON
BLOCK MODE	OFF
AUTO LF	OFF
BAUD RATE	9600
PARITY	NONE or O
ENQ/ACK	YES

LG200077_012

Remote Mode (ON)

The Remote mode setting is critical. It must be set to ON to send information to the computer. The opposite of Remote mode is Local mode. In Local mode, you can enter information and see it displayed on your terminal screen. However, it is not sent to the computer for processing.

Block Mode (OFF)

There are two ways to send data to the computer to be executed; either one line at a time or an entire screen all at once. Sending an entire screen is called Block mode. Block mode must be set to OFF.

Auto LF (OFF)

Linefeed is the process that moves the cursor to the first column of the next line when you press Return. This prevents the cursor from simply moving to the beginning of the line you just entered and writing over it. In Local mode (the opposite of Remote mode), Auto LF moves the cursor to the next line. In Remote mode, the computer automatically inserts linefeeds for you. Therefore, Auto LF must be set to OFF when Remote mode is ON.

Baud Rate

The baud rate is the rate--per second--at which bits are transmitted from the terminal to the computer and vice versa. The data exchange will work, no matter what the baud rate is. However, if you set the baud rate to a low setting, transmission will be very slow. At a baud rate of 300, for example, data sent from the computer to your terminal would appear on the screen very slowly.

The preset baud rate setting is 2400. You can set a higher baud rate, up to 19.2 K. However, if strange characters appear on your screen, lower the baud rate.

The recommended baud rate is 9600.

Parity (0 or NONE)

Parity is a form of error checking that calculates the number of odd and even digits that make up a character to the computer. If this calculation does not come out right, it indicates that something got lost in the transmission. Parity must be set to 0 (zero) or "none." Note that, on some terminals, this setting is combined with the "data bit" setting, as shown in Figure 2-8.

Enq/Ack (YES)

Enq/Ack are codes used to control the flow of data.

Checking and Changing Terminal Settings

This section explains how to use the function keys to define the terminal settings.

When you first turn on the terminal, small lighted squares appear at the bottom of the screen, as shown in Figure 2-5. Called "screen labels," these squares correspond to the function keys on the top row of the keyboard, which are labeled "f1" through "f8."

Key "f1," for example, affects the screen label above it, which displays "device control." When you press one of these keys, it changes the values that correspond to that key. The display in the screen label also changes to reflect the new values.

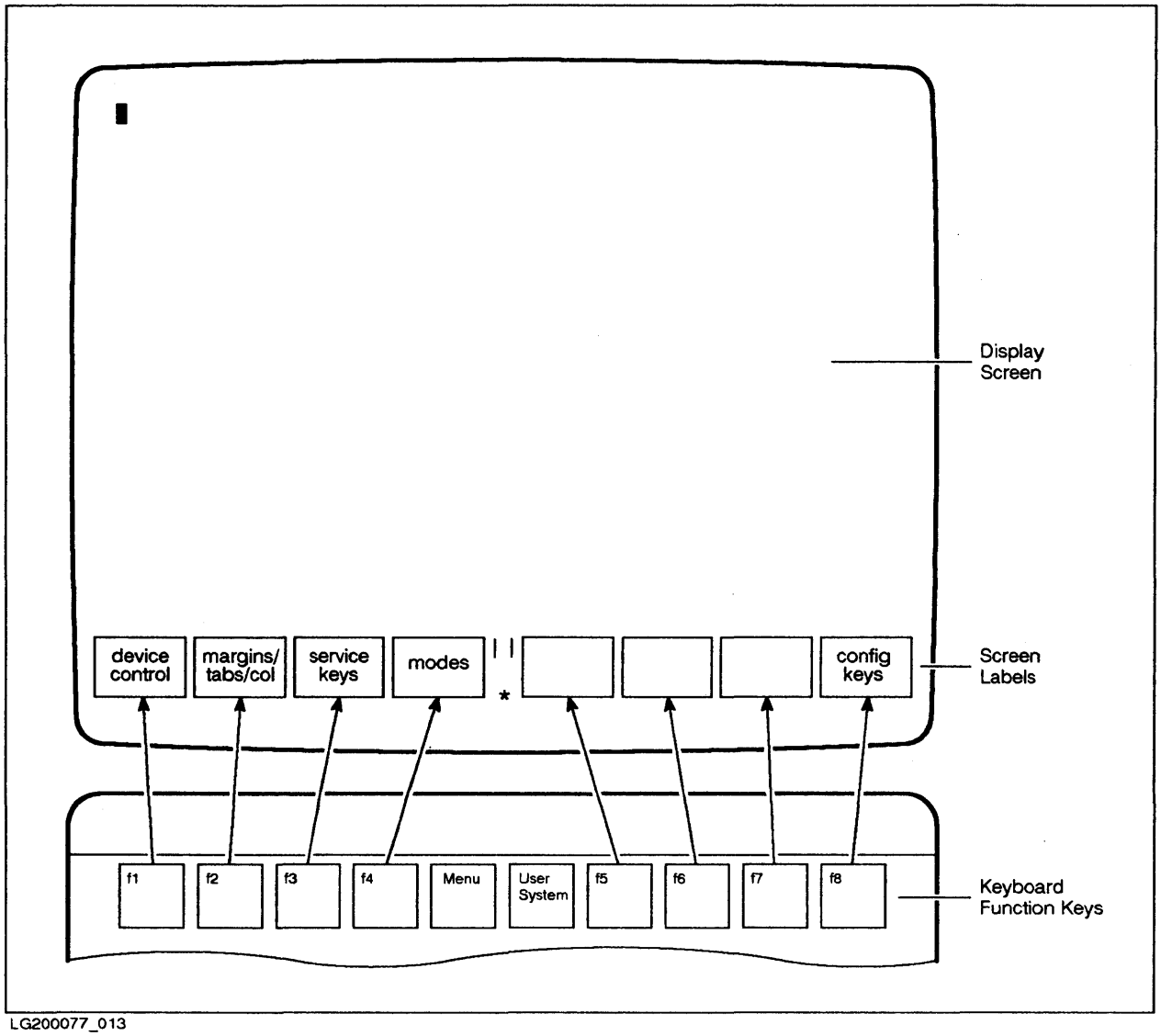


Figure 2-5. Default Terminal Settings

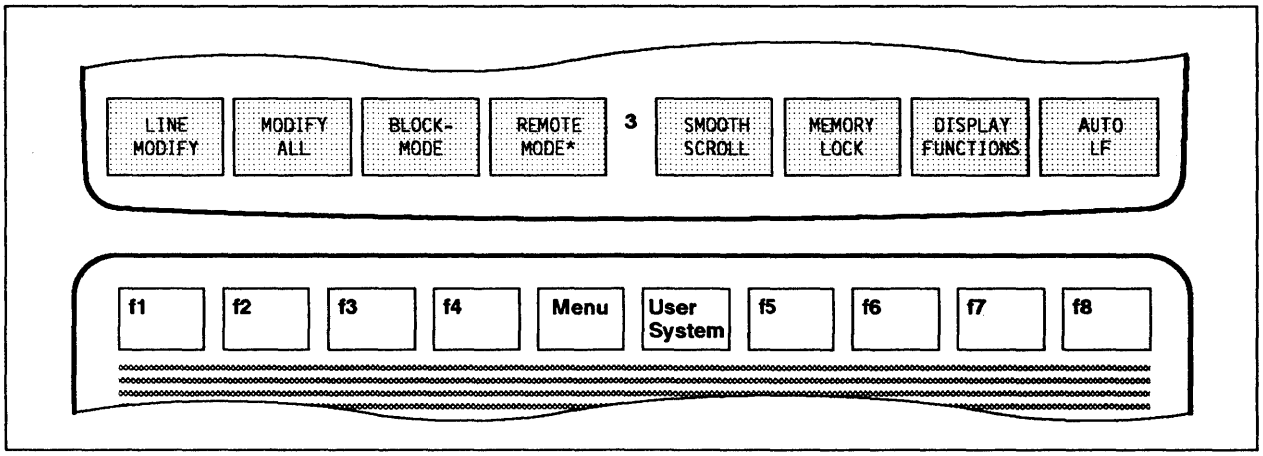
Checking Remote Mode, Block Mode, and Auto LF

When you first turn on the terminal, the function keys will appear as shown in Figure 2-5.

To check the terminal mode settings:

1. Press f4, which corresponds to the "modes" label in Figure 2-5.

   The labels change to look like Figure 2-6.



LG200077_014

Figure 2-6. Checking Terminal Modes Settings

2. Check the REMOTE MODE setting above [f4]. There should be an asterisk (*) beside the REMOTE MODE label. The asterisk means REMOTE MODE is ON.

   If there is no asterisk, press [f4]. The asterisk will appear in the square.

3. Check the BLOCK MODE setting above [f3]. There should be *no* asterisk (*) beside the BLOCK MODE label, indicating that BLOCK MODE is OFF.

   If there is an asterisk, press [f3]. The asterisk will disappear.

4. Check the AUTO LF setting above key [f8]. There should be *no* asterisk (*) beside the AUTO LF label, indicating that AUTO LF is OFF.

   If there is an asterisk, press [f8]. The asterisk will disappear.

Checking the Baud Rate, Parity, and Enq/Ack Settings

You can check and change the terminal's configuration settings. These settings define how devices like terminals, modems and printers communicate with the computer.

To check the configuration settings:

1. Press [User/System]. The screen will again appear as in Figure 2-5. (Skip this step if it already does.)

2. Press [f8], which corresponds to the "config keys" label.

   The labels change to look like Figure 2-7.

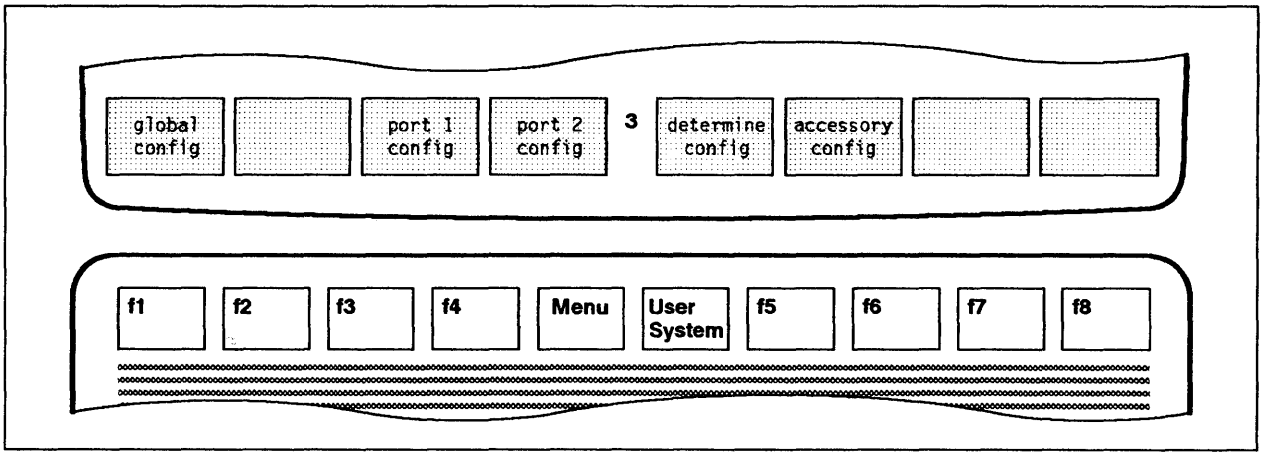


Figure 2-7. Checking Configuration Settings (1)

3. Press [f3], which corresponds to the "port 1 config" label.

   A display similar to Figure 2-8 appears. The cursor is in the Baud Rate Field.

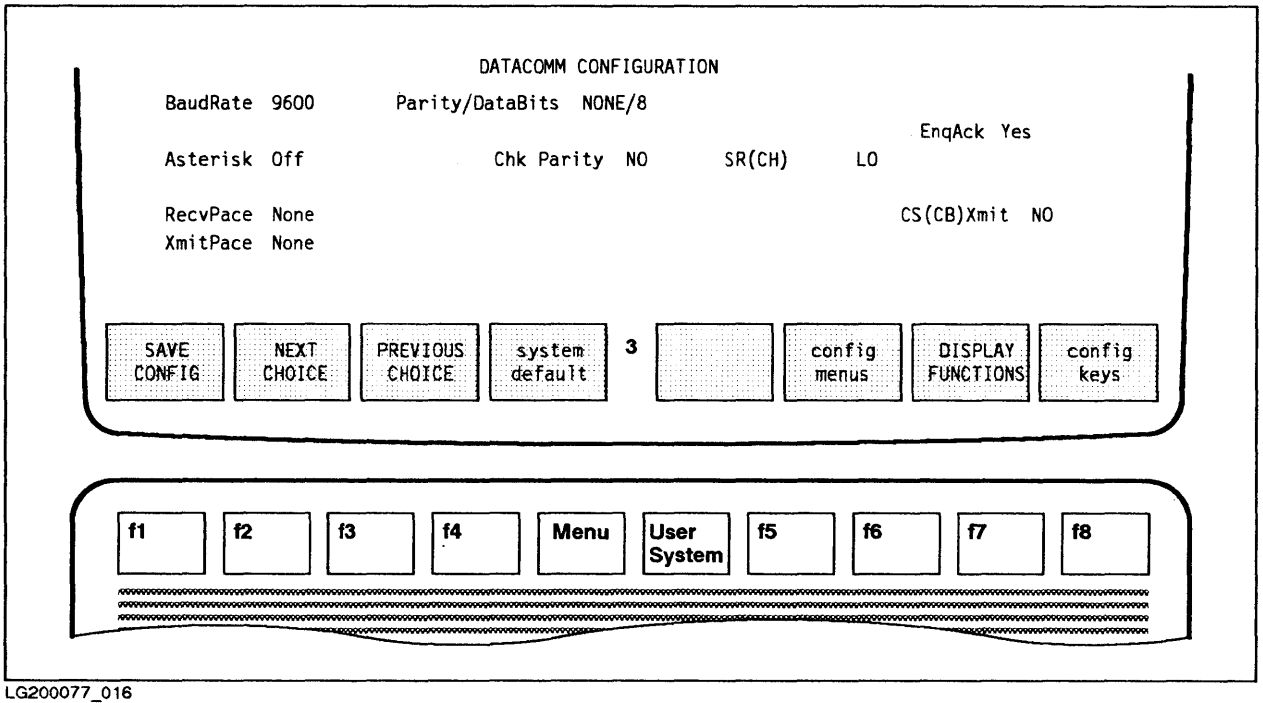


Figure 2-8. Checking Configuration Settings (2)

4. Press [f2] (NEXT CHOICE) repeatedly, until the desired baud rate appears in the Baud Rate field. (To return to a previous value, press [f3] (PREVIOUS CHOICE).

5. Press [Tab] until the cursor is in the field labeled "Parity/DataBits."

6. Press [f2] (NEXT CHOICE) until "None" appears in the Parity/DataBits field.

7. Press Tab until the the cursor is in the field labeled "EnqAck"

8. Press f2 (NEXT CHOICE) until "YES" appears in the EnqAck field.

9. Press f1 (SAVE CONFIG) to save these settings.

   The new settings take effect immediately and remain in effect until someone changes them.

The Keyboard

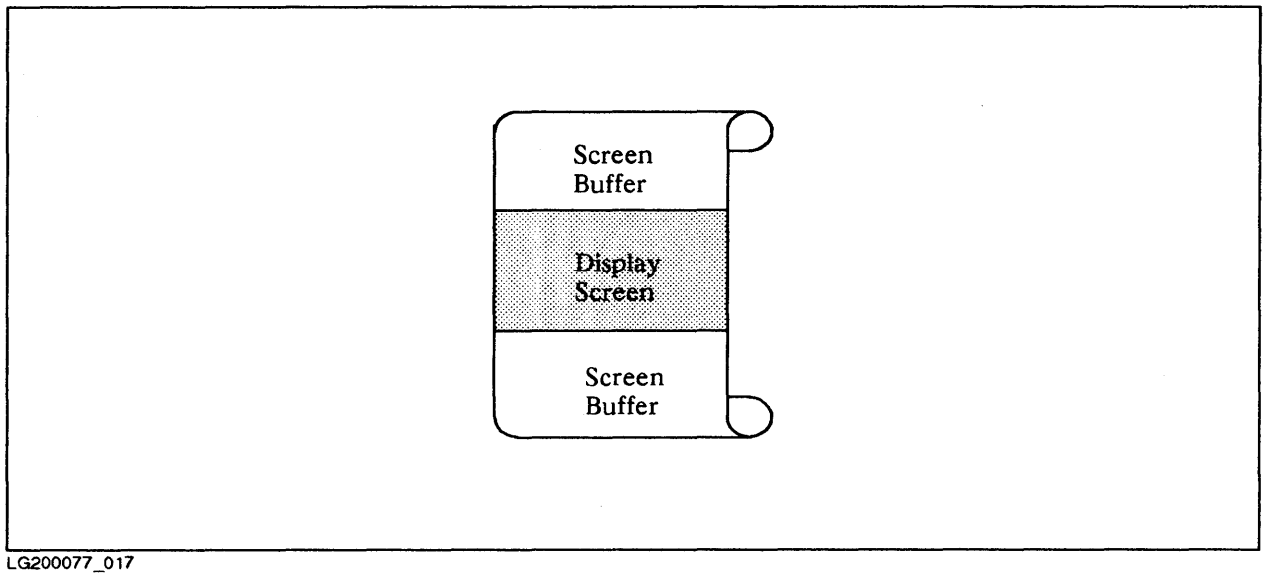
The keyboard lets you communicate with the computer. It looks similar to a typewriter keyboard and works in a similar way. Of course, information you enter is displayed on the screen instead of being printed on paper. Although each character is transmitted as it is entered, the information entered is processed only when you press Return.

In addition to the keys that work almost like typewriter keys, there are certain keys that are not used for entering data. Instead, they control how the terminal works. These "special" keys are discussed in more detail in this section.

Scrolling

As you enter, the screen eventually fills. After you enter the last line and press Return, the top line disappears to make room for a new line at the bottom of the screen. This process is called "scrolling." As you continue to enter, the screen continues to scroll.

Text that no longer appears on the screen is stored in the terminal's memory, which is called the "screen buffer." This concept is illustrated in Figure 2-9. However, there is a limit to the number of lines your terminal "remembers." For example, the HP 2392A keeps up to four screens in its memory (called the "screen buffer"). If the number of lines you enter exceeds the screen buffer, those lines are lost from terminal memory.



LG200077_017

Figure 2-9. Scrolling Data into the Screen Buffer

To display text scrolled into the screen buffer, you must set Block mode to on, as described in the previous section. You can then display this text.

To display text scrolled off the top of the screen:

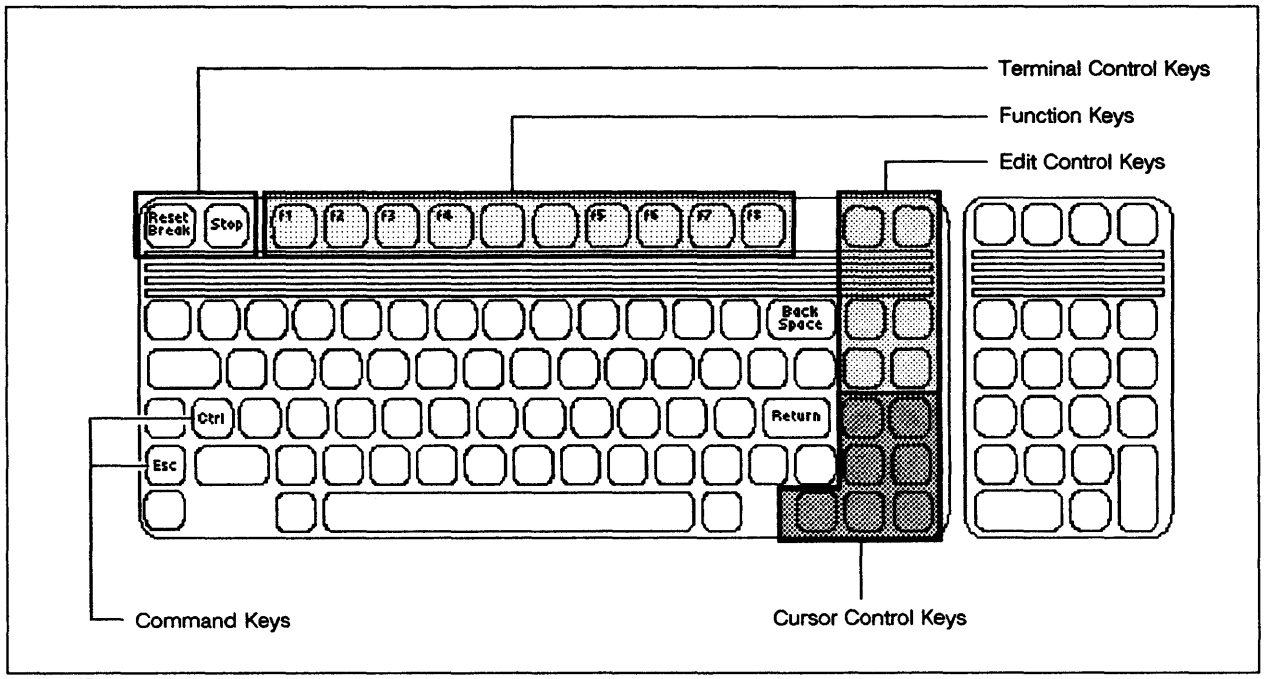
Press Shift and ▼ simultaneously.

To display text scrolled off the bottom of the screen:

Press Shift and ▲ simultaneously.

Comparison with Typewriter Keyboards

Figure 2-10 shows the HP 2392A's keyboard. (These keys may be marked differently on other terminal keyboards and they may be in different locations).



LG200077_018

Figure 2-10. HP 2392A Standard USASCII Keyboard

Most of these keys function similarly to a typewriter's keys, including the characters, the numbers, the punctuation marks, and special symbols. When you press one of these keys, the corresponding symbol appears on the screen. When you press [Return], that character or group of characters is processed by the computer. In addition, the keys shown in Table 2-2 work the way they do with typewriters.

Table 2-2. Keys that Function Like Typewriter Keys

Key	Function
Shift	Shifts to upper case letters or special characters.
Return	Moves the cursor to the beginning of the next line.
Space bar	Puts in a blank character.
Back Space	Moves the cursor backwards one space.
Tab	Skips to predefined columns in many programs.

LG200077_019

The "Special" Keys

The keys shown in Table 2-3 have special functions. Some keys work by themselves, others work together with other keys. These keys can interrupt programs, send data to the computer, control the screen display, display entire words with the touch of a single key, and more. This section describes those special keys used by general users.

Many of these keys trigger codes that send commands to the computer. You can enter these commands at the keyboard or incorporate them into programs, in any programming language. To find out how to perform some of these functions programmatically, refer to your terminal's documentation.

The special keys are summarized in Table 2-3:

Table 2-3. "Special" Keys

Key Category	Function
Terminal control	Interrupt communications with the computer and reset the terminal.
Edit control	Insert and delete characters and lines.
Cursor control	Control cursor movement.
Command keys	Control the terminal when used with other keys.
Function keys	Perform different functions depending on what is displayed in squares at bottom of screen. Can also be programmed to display phrases.

LG200077_020

The Terminal Control Keys

These keys can interrupt communication between the terminal and the computer and reset the terminal.

Break/Reset has several functions. It may be used when the keyboard does not respond, or to temporarily interrupt or abort user programs or commands. These procedures are discussed in Chapter 8, "Working with Sessions," and in the *HP 2392A Display Terminal Reference Manual* (02390-90001).

Return tells MPE to process the information entered. After you press this key, the cursor moves to the next line. Note that, in some terminals, the Enter key performs the same function as the Return key. In this case, it is also in the location normally reserved for the Return key.

Enter sends an entire screen of information to the computer when the terminal is in Block mode. Many programs tell you when to press this key.

The Edit Control Keys

These keys let you edit text displayed on the screen. They are used mostly for applications such as filling in forms on the screen. If you want to permanently save changes to files, it is better to use the EDITOR subsystem or another wordprocessing program.

NOTE

The edit or cursor control keys work as described only in Block mode (also called "full screen" mode). Otherwise, use the Space bar or the Back Space key to move the cursor.

Clear display erases the screen below the cursor.

Insert line displays a new, blank line above the line that contains the cursor.

Insert char lets you insert one or more characters, starting at the cursor.

Delete line deletes the line that contains the cursor.

Delete char lets you delete one or more characters, starting at the cursor.

The Cursor Control Keys

The cursor control keys move the cursor around the screen and into the screen buffer (Figure 2-9). If you press these keys once, they move one character or line at a time. If you hold them down, they move across the screen very quickly and then start over. (Try it to see how it works.) You can also move the cursor to the beginning or end of the screen buffer, or forward or backwards one screen at a time.

NOTE

The edit or cursor control keys work as described only in Block mode (also called "full screen" mode). Otherwise, use the Space bar or the Back Space key to move the cursor.

[▶] moves the cursor to the right, character by character.

[◀] moves the cursor to the left, character by character.

[▼] moves the cursor down, line by line, to the bottom of the screen display.

[▲] moves the cursor up, line by line, to the top of the screen display.

[Shift] and [▼] together scroll the text down one line to display the first line in the top screen buffer.

[Shift] and [▲] together scroll the text up one line to display the first line in the bottom screen buffer.

[◸] moves the cursor to the first character in the first line in the screen buffer. This position is often called the "home" position.

[Shift] and [◸] together move the cursor to the beginning of the last line in the screen buffer. This position is often called the "end" position.

[Prev] moves the cursor back approximately 24 lines in the screen buffer. This is often called the "previous page" or "PG Up."

[Next] moves the cursor forward approximately 24 lines in the screen buffer. This is often called the "next page" or "PG Down."

The Command Keys

[ESC] and [CTRL] are command keys that can be used with another key to trigger command sequences that control the terminal. You can enter these sequences at the keyboard or include them in programs to control the terminal automatically. For example, you can determine whether or not characters appear on the screen as they are entered.

Using Command Sequences

The [CTRL] and the [ESC] keys differ technically and they are used in a slightly different way.

To control the terminal with a control sequence:

> Press [CTRL] and hold it down while pressing the other key in the sequence. The other key can either be upper or lower case.

To control the terminal with an escape sequence:

> Press [ESC], release it, and press the other key in the sequence. Note that the other key in the sequence must be upper case.

This manual only discusses the most commonly used command sequences, which are summarized in Table 2-4 and described in more detail below. Note that not all sequences work with all programs.

Table 2-4. Summary of Command Sequences

Command Sequence	Function
ESC ;	Turns off screen echo (characters invisible).
ESC :	Turns on screen echo (characters visible).
CTRL S	Stops a screen listing.
CTRL Q	Resumes a screen listing.
CTRL Y	Interrupts some subsystems to display main subsystem prompt.
ESC J	Clears the display of text (works only in certain programs).
CTRL G	Causes the terminal to beep.

LG200077_021

ESC ; turns off the "screen echo". As a result, the characters you enter no longer appear on the screen. However, information is still sent to the computer.

Many programs turn the screen echo on and off automatically as needed. However, if your text mysteriously disappears, you may have pressed this sequence by mistake.

ESC : turns the screen echo back on. If you entered ESC ; by mistake, this sequence makes your text reappear.

CTRL S interrupts a screen listing. For example, if you entered a command that lists all the files in your account, they will scroll onto the screen too quickly to be read. To find a particular file, you have to stop the listing to locate that file.

CTRL Q resumes a screen listing stopped with the previous command. Once you find what you were looking for, you can again start the listing where it left off.

NOTE

If your terminal has a [Stop] key, use it instead of the command sequences [CTRL] [S] and [CTRL] [Q].

[CTRL] [Y] works in certain programs to completely stop a screen listing or program. To run that listing or program again, it has to be started from the beginning. This command is useful when you realize you requested a listing you do not really want.

[ESC] [J] clears the screen display in certain programs.

[CTRL] [G] causes terminal to beep. This may be useful to get a user's or Operator's attention when you send a message.

The Function Keys

As you know, you can check and change the terminal settings by pressing the function keys that correspond to the displays in the screen labels at the bottom of the screen.

The function keys can also be great timesavers if you enter the same commands, words, or sentences throughout your workday. These keys can act like storage areas that hold up to one line of data (called a "character string"). By "programming" these keys, you can then display that character string by simply pressing the key that holds it. For example, if you have to enter the date many times each day, you can program one of the function keys to insert the date whenever you press it.

The function keys will "remember" the programmed character strings until you reprogram them or turn off the terminal's power.

Programming the Function Keys

Programming and using the function keys consists of three basic steps:

- Displaying the "label" screen
- Entering information for each function key, including the identifying word that appears in the screen label and the information you want to display when you press the corresponding key.
- Displaying the screen labels to use the function keys.

You can also define the function keys programmatically. For instructions, refer to the *HP 2392A Display Terminal Reference Manual* (02390-90001).

To program the function keys:

1. Display the label screen by pressing CTRL and Menu simultaneously.

   The label screen, which resembles Figure 2-11, appears. The cursor is in the field labeled "T".

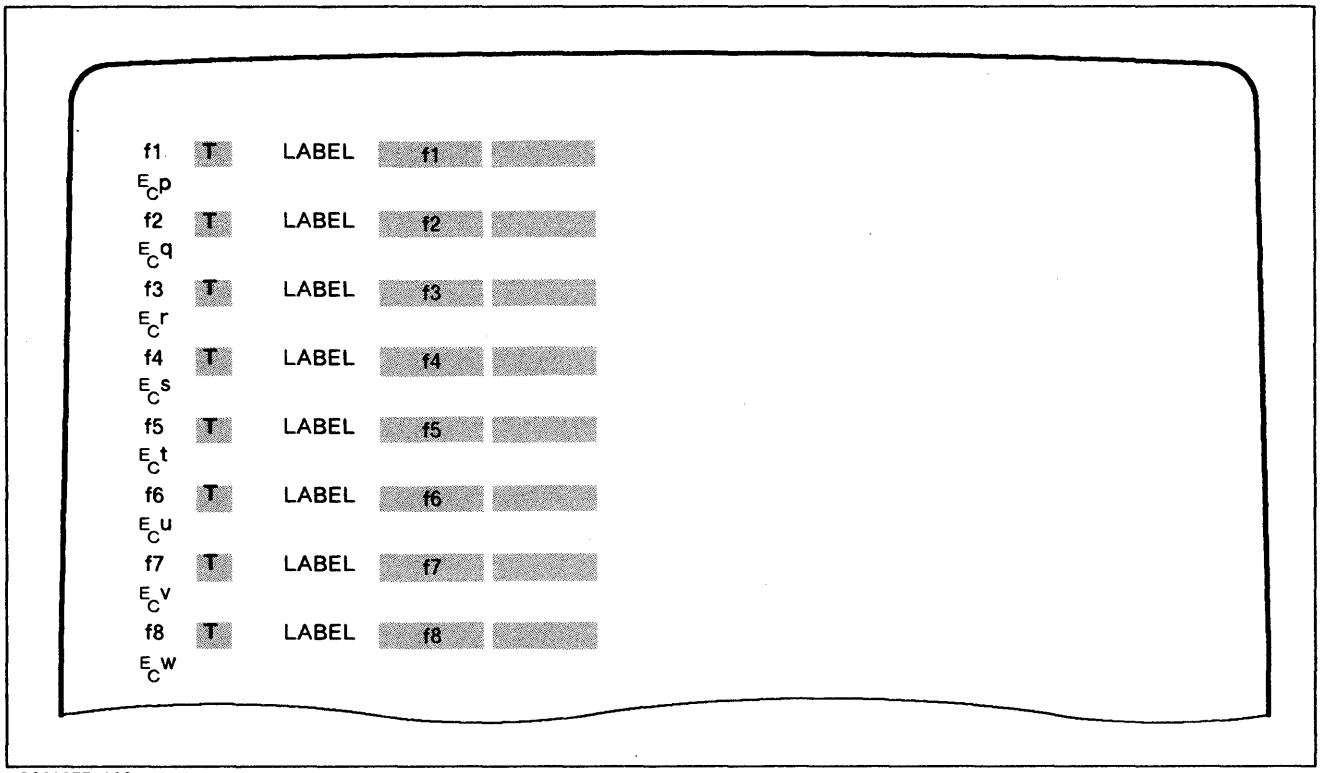


Figure 2-11. The Function Key Label Screen

2. Enter information. To move forward between fields, press Tab. To move backward between fields, press SHIFT and Tab simultaneously.

   a. Press Tab to move to the field labeled "f1."

   b. Enter the label (up to eight characters) that you want to appear in the upper half of the screen label that corresponds to function key f1.

   c. Press Tab to move to the next field.

   d. Enter the label (up to eight characters) that you want to appear in the lower half of the screen label that corresponds to function key f1.

   e. Press Tab to move to the field labeled "EC p".

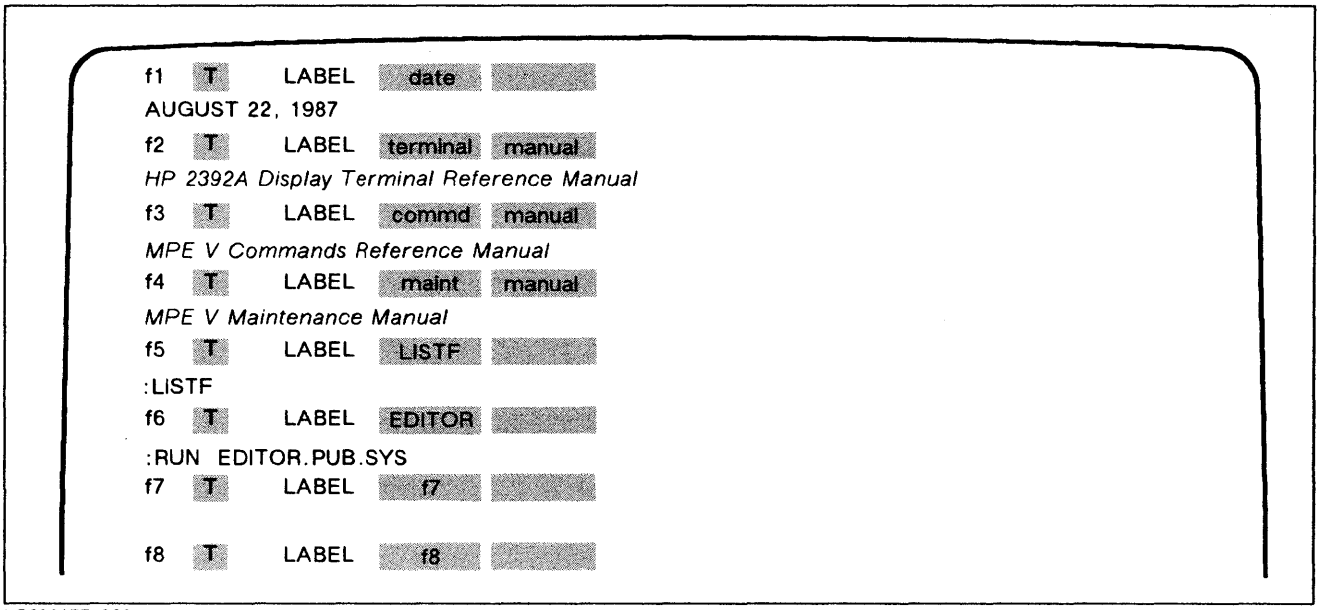
   f. Enter the characters (up to 80 characters) that you want to display when you press f1.

g. Repeat steps a. through f. to program any other function keys.

h. Press [User/System] when you finish entering information.

Figure 2-12 shows how to fill in the label screen to program the function keys.

Key [f1] will display the date, [f2], [f3], and [f4] frequently used book titles. Keys [f5] and [f6] will execute commands when pressed. Keys [f7] and [f8] are not programmed. Figure 2-13 shows how the lighted squares would appear on the screen.



LG200077_023

Figure 2-12. Programming the Function Keys

Using the Programmed Function Keys

When you finish programming the function keys, you are ready to use them.

To use the programmed function keys:

1. Press [CTRL] and [User/System] simultaneously to display the labels in the lighted squares, as shown in Figure 2-13. You are now ready to use these key as programmed.
2. Press the function key that contains the information you want to display.
3. When no longer needed, press [User/System] again to redisplay the function keys with their original labels (Figure 2-5).

Figure 2-13 shows how the keys programmed in Figure 2-12 would appear on the screen.

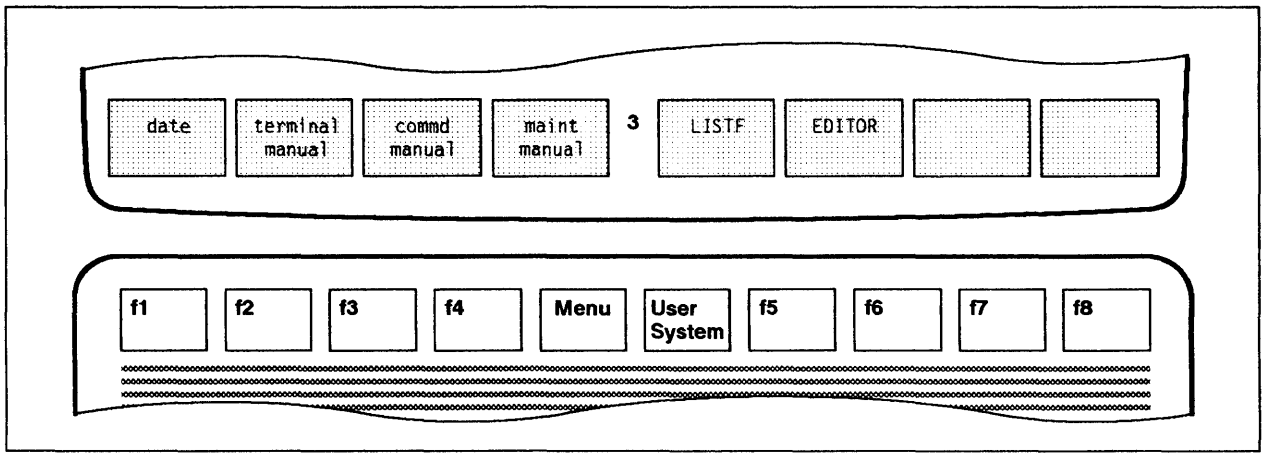


Figure 2-13. Using the Programmed Function Keys

Printing What Is on the Screen

The terminal screen shows what you entered and the system's response. If a printer is connected to your terminal, you can print this information as it appears on the screen. Printing what is on the screen is also known as a "screen dump."

Even if you do not have a printer at your workstation, MPE automatically prints the output of jobs. Many commands also have an option that let you print listings on the printer instead of displaying them on the screen.

Printing screen dumps and listings could be helpful in several situations. For example, if you requested a long listing, such as a display of all the files in your account, the information would scroll by so quickly you could barely read it. Of course, you could use CTRL S to stop and CTRL Q to restart the listing as often as necessary. However, a printout of the entire listing would be much easier to use.

Printing Screen Dumps at Your Workstation

To print screen dumps at your workstation, a printer must be connected to your terminal with the printer connection cable. In addition, certain configuration settings also must be correct to allow the terminal and the printer to communicate.

To connect the printer cable to the terminal:

1. Turn off the terminal.
2. Plug one end of the printer cable into the terminal. If necessary, tighten the screws. Refer to Figure 2-14 to locate the printer cable connection on the HP 2392A terminal. If you have a different terminal, refer to its documentation or ask your manager.

3. Plug the other end of the printer cable into the printer. If necessary, check the printer's documentation or ask for help.

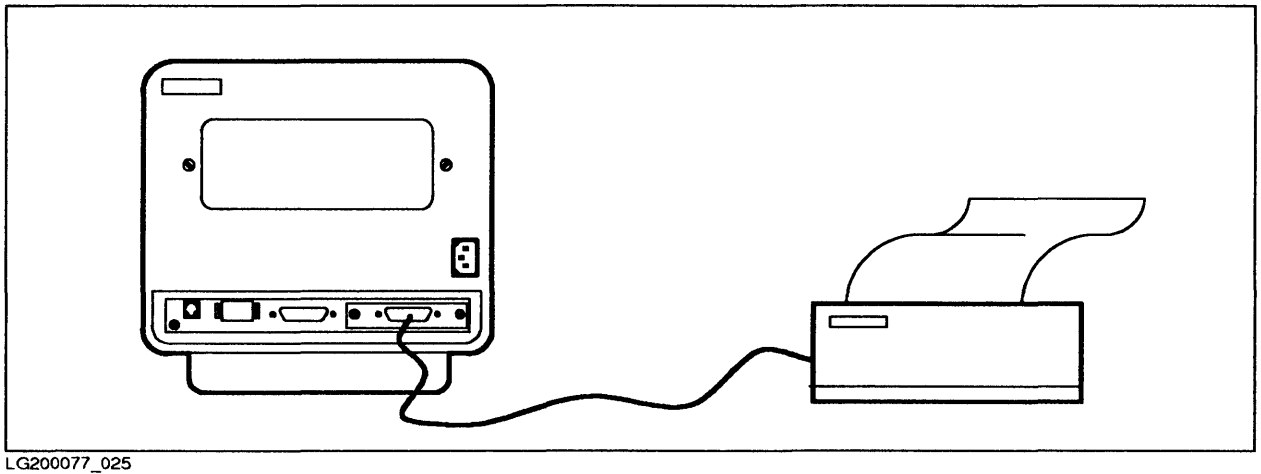

LG200077_025

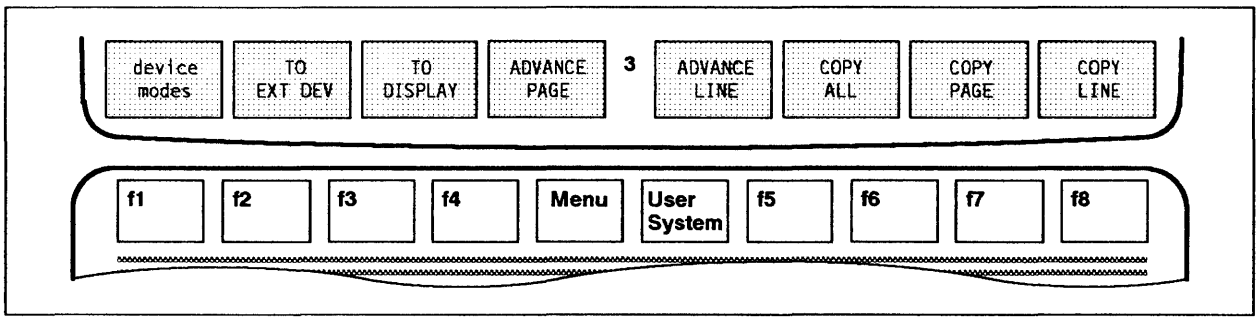
Figure 2-14. The Printer Cable Connection for the HP 3292A Terminal

To print a screen dump on your workstation printer:

1. Press [User/System]. The screen will appear as in Figure 2-5. (Skip this step if it already does.)

2. Press [f1] ("device control").

   The screen labels change to look like Figure 2-15.



LG200077_026

Figure 2-15. Printing a Screen Dump (1)

3. Press [f1] ("device modes").

   The screen labels change to look like Figure 2-16.

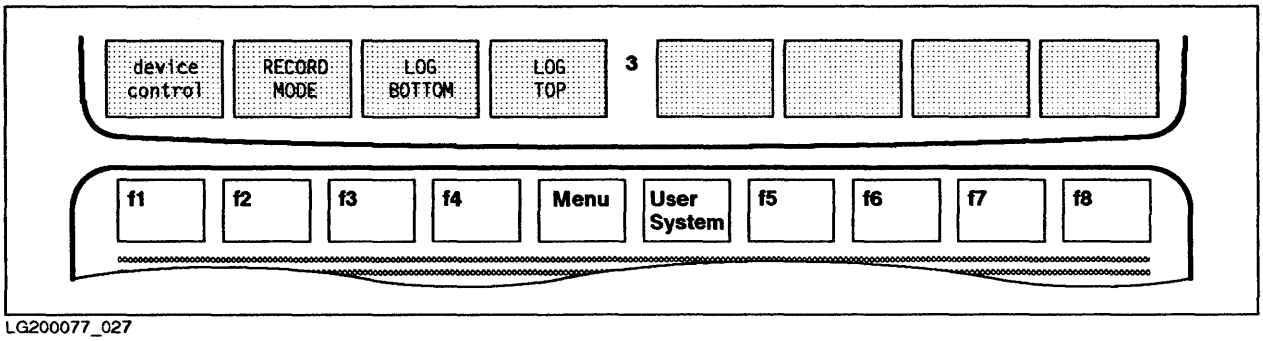


Figure 2-16. Printing a Screen Dump (2)

4. Press [f3] ("LOG BOTTOM") and [Return].

   An asterisk (*) appears in the label and the printer starts to print anything you enter, as well as any responses from the computer.

5. To turn off the printing, press [f3] and [Return] again.

   An asterisk (*) disappears from the "LOG BOTTOM" label and the printer stops printing.

Printing Listings on the System Printer

If you do not have a printer at your work station, you can still print any information generated by an MPE command (also called a "listing") on the system printer. The concepts that relate to this procedure are discussed in Chapter 9, "Jobs".

To print a listing:

1. Enter `STREAM` at the colon (:) prompt.

   MPE displays a chevron (>) prompt.

2. Enter `!JOB`, followed by the user logon (including passwords, if any).

   ```
   >!JOB username.accountname
   ```

3. Enter the command for which you want to print a listing, preceded by an exclamation mark (!)

   ```
   >!COMMAND
   ```

4. Enter `!EOJ`

   ```
   >!EOJ
   ```

5. Enter a colon (:).

   MPE displays the colon (:) prompt.

Example 2-1 shows how the user MGR in the account PAT could print a listing of the :SHOWCATALOG command on the line printer.

```
:STREAM
>!JOB MGR.PAT
>!SHOWCATALOG
>!EOJ
>:
:
```

LG200077_028

Example 2-1. Printing a Listing on the Line Printer

3

Introduction to Commands

Chapter Overview

This chapter introduces you to the commands that control the MPE operating system. It includes the following information:

- Basic concepts
- The parts of a command
- How to use the *MPE V Commands Reference Manual* (32033-90006)
- How to use the HELP facility
- An overview of commonly used commands

Introduction

A command is a word that tells MPE to do something. This usually means running programs, managing files, managing jobs and sessions, sending messages, and monitoring the system.

For example, the :HELLO command, followed by the user's name and account, tells MPE to initiate a session. In response, MPE searches its directory for the user and account specified. If it finds them, it starts a session for that user.

Example:

```
:HELLO MGR.PAT
```

If MPE finds the user (MGR) and the account (PAT) specified, it connects that user to the HP 3000.

Within MPE, there is a program called the "Command Interpreter," or "CI" for short. When you enter a command, this program checks the spelling and "syntax," which includes the punctuation and the order of the different parts of the command. If you make a mistake, the terminal displays a message known as a "CI error." In most cases, this message includes information that should help to fix the error.

MPE understands and obeys hundreds of commands. Luckily, you can do a lot of work with relatively few commands. Once you understand how to use the *MPE V Commands Reference Manual* (32033-90006), you can look up the details for any command. You can also display the same information on your terminal screen by using the HELP facility.

NOTE

This chapter includes step-by-step instructions for several tasks. To avoid repetition, assume the following:

1. At each step, you enter information at the colon (:) prompt, unless specified otherwise.

2. After entering the information, you press Return.

Anatomy of a Command

A command has at least two parts: the prompt and the command name. It may or may not have parameters. The way these parts are put together is known as a command's "syntax."

Figure 3-1 shows the syntax for a command with parameters (:HELLO) and without parameters (:BYE).

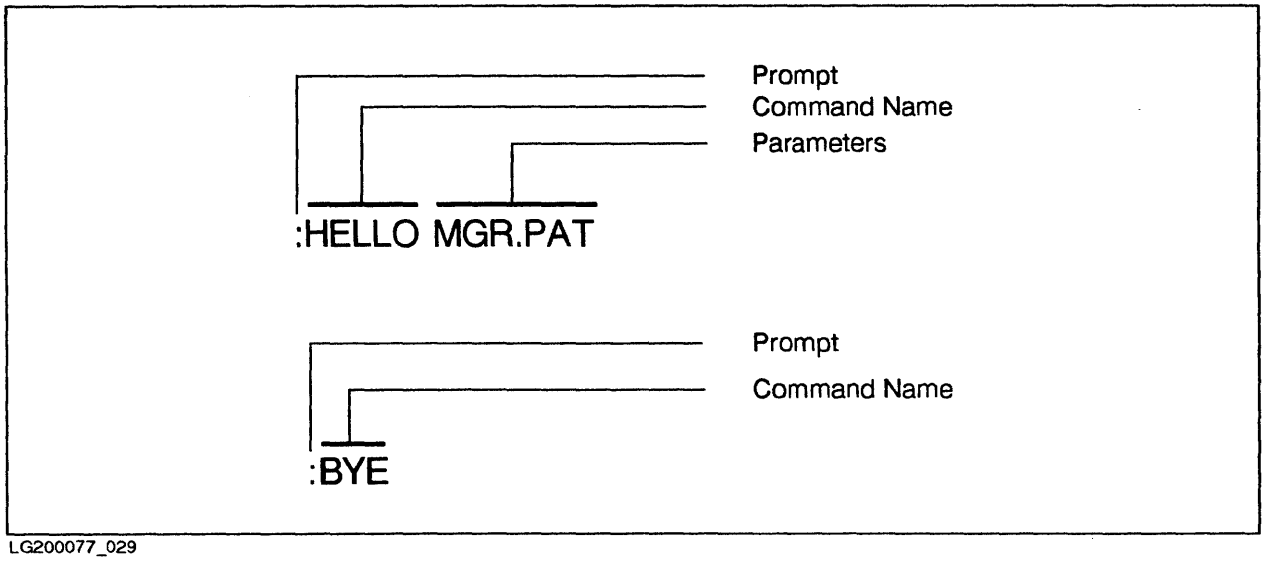


Figure 3-1. Anatomy of a Command

Prompt

After you successfully log on, MPE displays the : (colon) character, which is called the "prompt." This means MPE is ready to receive a command. The : (colon) prompt reappears after you enter a command correctly, indicating that the system executed the command and is ready for the next one. For jobs, MPE does not display the : (colon) prompt. Instead, you must enter a character that serves as a prompt before each command in your job file, as explained in more detail in Chapter 9, "Jobs."

When you enter one of MPE's subsystems or utilities, a different prompt may appear to let you know that a new range of commands is available. Common subsystem prompts are the / (slash) and > (chevron) symbols. For example, the text processing subsystem EDITOR uses the / (slash) prompt, indicating that you can execute EDITOR commands. However, you can still execute MPE commands from most subsystems by entering a : (colon) in front of the desired MPE command. Subsystems and utilities are explained in more detail in Chapter 5, "Utilities and Subsystems."

Command Name

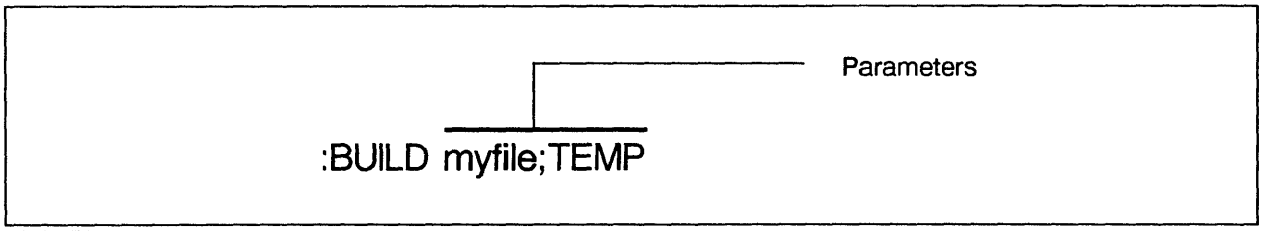
The command name is the main part of the command that tells the computer "do this." When you see a command name in a Hewlett-Packard manual, that command is always spelled in upper case letters (caps). You can enter the command in upper case or lower case letters, but you must spell it correctly.

Parameters

Parameters are words that affect how a command works. They are usually separated from the command name by a blank or a semicolon, depending on the command. Many commands have no parameters because no options are necessary or allowed. The `:BYE` command, for example, simply logs off the user who enters it.

In commands that have parameters, the command name specifies "do this," while the parameters go on to say "to that" and "in this way."

In Figure 3-2, the command name `BUILD` tells MPE to create a file. The parameters go on to say "call it myfile" and "make it temporary."



LG200077_030

Figure 3-2. Parameters

Required Parameters

If the command name says "do this," the required parameter specifies "to that." "That" may be a file, a program, or a user; thus defining which of many possible files, programs, or users the command will apply to. Without this information, MPE cannot execute the command. In Figure 3-2, the required parameter was the name of the file to be created (`myfile`).

Another example is the `:RUN` command. Unless you tell MPE *which* program you want it to run, it cannot execute the command. In this case, the name of the program is the required parameter.

When looking up commands in the *MPE V Commands Reference Manual* (32033-90006), you will recognize required parameters because they are *not* surrounded by brackets.

Optional Parameters

So far, the command name and the required parameter have specified "do this to that." Optional parameters go on to say "in this way." If you do not specify optional parameters, MPE automatically assigns values for those parameters. Whenever you leave a decision up to the computer, that decision is called the "default." MPE has default values for all optional parameters, which are the values most commonly used.

In Figure 3-2, the default value was to make the file permanent, since that is what most users want. However, since `TEMP` was specified as an optional parameter, MPE creates a temporary file instead.

When looking up commands in the *MPE V Commands Reference Manual* (32033-90006), you will recognize optional parameters because they are surrounded by brackets.

Positional and Keyword Parameters

MPE interprets both required and optional parameters in two ways: by their position within the command or by their association with a keyword. These parameters are called "positional" or "keyword," respectively.

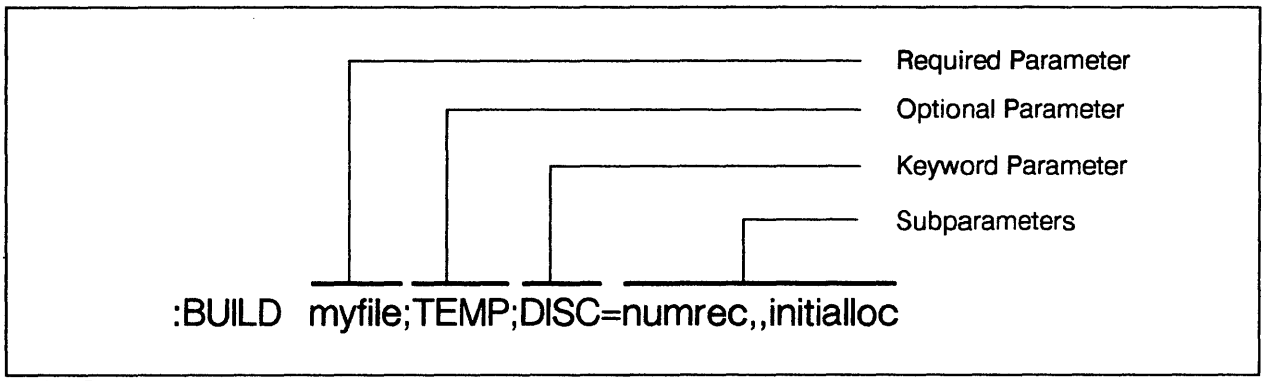
Keywords define the subject. They say, in effect, "what are we talking about?" Keywords may be associated with a single value or several subparameters, each with a value. Any values or subparameters that follow the keyword further describe that subject, saying "what about it?" In Figure 3-3, the subparameters associated with the DISC= keyword are `numrec, numextents`, and `initialloc.`

When looking up commands, you will recognize keyword parameters because they are preceded by a semicolon, because they appear in all caps, and because they are associated with one or more values. You will recognize positional parameters because they are preceded by a comma or slash (for passwords) and because they appear in italics.

Figure 3-3 illustrates the concepts described so far.

- A required parameter (`myfile`), which also happens to be a positional parameter
- An optional parameter (TEMP)
- A keyword parameter (DISC=)
- Subparameters (`numrec` and `initialloc`) associated with the DISC= parameter.

This command creates a file called `myfile`. The optional parameter TEMP tells MPE to make the file temporary, while the subparameters associated with the keyword parameter DISC= define how disc space is allocated for that file. Note that the subparameter "numextents" is omitted and that there is a comma in its place. (If these values mean nothing to you, don't worry. Remember that they are optional: if you do not specify optional parameters, the computer automatically assigns default values. These values work just fine unless you have special needs.)



LG200077_031

Figure 3-3. Relationship between Positional, Keyword, and Subparameters

Summary: Characteristics of Parameters

Here are some characteristics of positional, keyword, and subparameters:

Positional Parameters

- Positional parameters are separated by commas (,).
- If used, positional parameters must be defined in the order listed in the command's syntax, as specified in the *MPE V Commands Reference Manual* (32033-90006).

Keyword Parameters

- Keywords are preceded by a semicolon (;).
- Keywords are followed by an equals (=) sign and either a single value or several subparameters.
- If a command contains more than one keyword, they may appear in any order.

Subparameters

- Subparameters associated with a keyword have the same characteristics as positional parameters.
- In a group of subparameters, you must insert a place-holding comma for any omitted subparameters.

Delimiters

Delimiters are the punctuation marks that separate parameters. They include:

- Blanks (or semicolons), which follow the command name.

- Commas, which separate positional parameters.
- "Place-holding" commas, which indicate that a subparameter has been left out.
- Semicolons, which precede keyword parameters.
- Slashes, which precede passwords.
- Equal signs (=), which separate the keyword parameter from the subparameters that further define it.

Figure 3-4 is identical to Figure 3-3, except that the focus is on the delimiters.

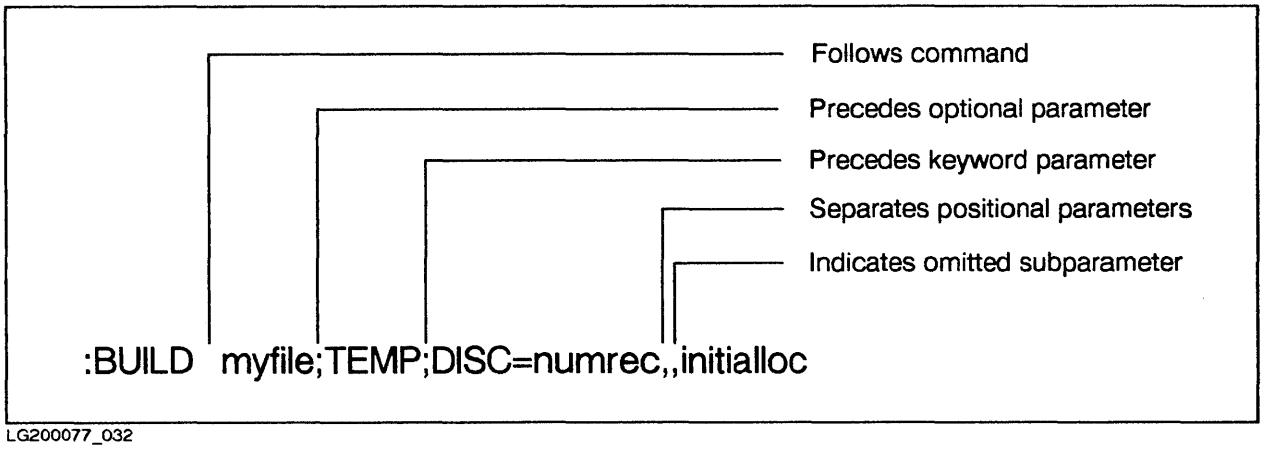


Figure 3-4. Delimiters

The Ampersand Continuation Character

Some commands have many possible parameters. To specify more parameters than will fit on one line, you can add an ampersand (&) character immediately before or after a delimiter. This allows you to enter additional parameters on the next line before sending the command to the computer. It may also make the line more readable.

When working in a session, a colon prompt (:) appears at the beginning of the new line in response to the ampersand (&) character. In a job, however, you must enter a new prompt at the beginning of each line.

Figure 3-5 illustrates the use of the ampersand (&) character.

```
:BUILD myfile;REC = 128,1,F,ASCII;&
:DEV = DISC;DISC = 128,8,1
```

LG200077_033

Figure 3-5. The Ampersand Continuation Character

Reading Documentation

Once you recognize the different parts of a command, you can use the *MPE V Commands Reference Manual* (32033-90006). This section summarizes the information in this manual.

In addition, the *MPE V Quick Reference Guide* (32033-90023) provides experienced users with a quick reference to command syntax, utilities, and more.

As you can see, commands can either be short and simple or relatively complex, depending on whether you define optional parameters.

Reading the Commands Reference Manual

The *MPE V Commands Reference Manual* (32033-90006) is organized alphabetically by command. Each command entry has the following components: Command Name, Syntax, Parameters, Use, Operation, Example, and Additional Discussion.

As you read this section, turn to the :HELLO command.

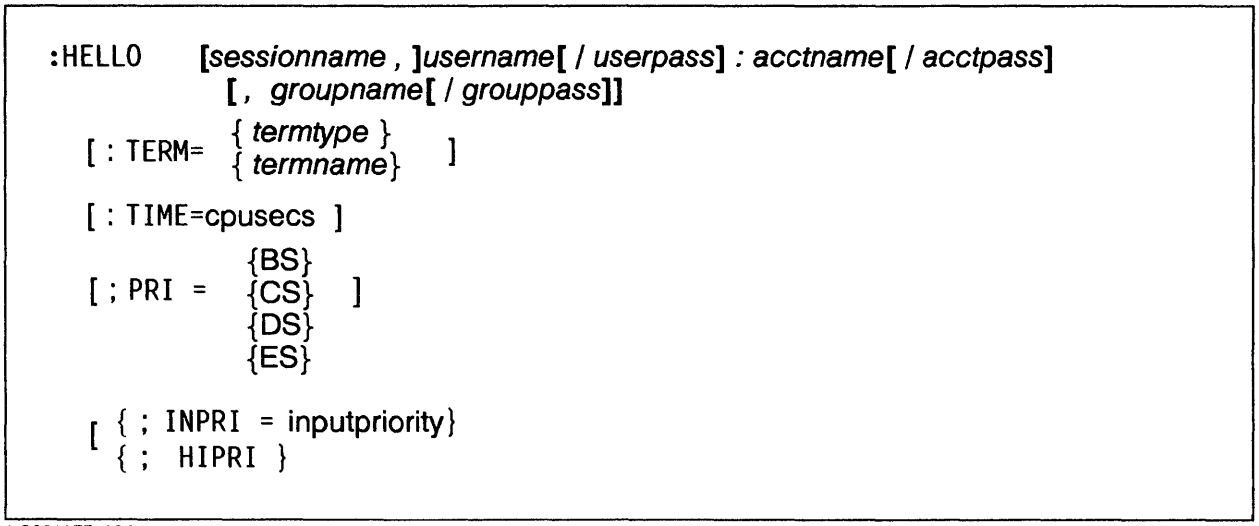
Command Name

The command name at the top of the page is followed by a brief description of what the command does.

Syntax

The syntax is shown in the "syntax box," which shows the command name, followed by the parameters, if any. Some commands have so many different parameters, with several options for each parameter, that the result looks pretty confusing.

Take :HELLO, for example. It is a simple command; you have probably used it without a second thought. However, if you look it up, the syntax box looks like Figure 3-6.



LG200077_034

Figure 3-6. :HELLO Command Syntax

In this command, the required parameters are *username* and *accountname* (they are not surrounded by brackets.) All other parameters *are* surrounded by either [ ] (brackets) or { } (braces), indicating that they are optional.

NOTE

Remember: in most cases, all you need for a command to work is the command name and any required parameters.

Parameters

The "Parameters" section provides additional information about each of the parameters listed in the syntax box. This includes any special rules, such as the length of the name, the default value, and what punctuation marks (delimiters) must be associated with various parameters.

In the :HELLO command, for example, one of the optional parameters in the first line is *acctpass*. This section tells you that *acctpass* defines the account password, that this password must contain from one to eight alphanumeric characters, and that it must be preceded by a slash.

Use

The "Use" section explains whether you can issue a command from a session, job, or program and whether you can stop its execution with the Break key. In addition, it defines the capabilities necessary to execute this command, which are discussed in more detail in Chapter 6,

"Basic Account Structure". For example, only users with SM (System Manager) capability can execute the command that deletes (purges) accounts. In addition, certain commands can only be executed at the terminal designated as the System Console.

Operation

The "Operation" section explains in more detail what the command actually does, as well as anything else you may need to know about the command.

Examples

The examples, hopefully, clarify everything else. The underlined text in the *MPE V Commands Reference Manual* (32033-90006) shows what the user enters. The text that follows the underlined area shows the computer's response.

Additional Discussion

The "Additional Discussion" section refers you to other relevant documentation.

Summary: Reading the Commands Reference Manual

The *MPE V Commands Reference Manual* (32033-90006) includes a page titled "Conventions Used in this Manual." Here are some key points to remember:

- Upper case letters indicate that you must enter the information exactly as shown.
- Except for keyword parameters, parameters appear in italics.
- Required parameters are *not* surrounded by [ ] (brackets) or { } (braces). You must define values for these parameters.
- Optional parameters *are* surrounded by [ ] (brackets) or { } (braces), indicating that defining values is optional. Optional parameters can be positional or keyword parameters.
- Positional parameters are preceded by commas. When you specify positional parameters, do so in the order specified in the command.
- Keyword parameters are preceded by semicolons and appear in all caps.
- Subparameters may be associated with keywords. If there is more than one subparameter, you must list them in the order specified in the command. If you omit a subparameter, insert a place-holding comma in its place.
- If several parameters in brackets are stacked on top of each other, you *can* choose any one of those parameters. If they are also surrounded by braces, you *must* choose one of them. In Figure 3-6, for example, if you specify the keyword parameter TERM, you must also specify

either the *termtype* or *termname*. If you specify the keyword parameter PRI, you must also specify either BS, CS, DS, or ES.

- You must use the shaded punctuation marks (delimiters) if you use the parameters they precede. In Figure 3-6, note the following delimiters: slashes (/) before passwords, semicolons (;) before keyword parameters, and a comma (,) preceding the positional parameter *groupname*.

This section is followed by a chapter titled "Introduction to Commands." As you become an experienced user, be sure you read this information.

Reading the Quick Reference Guide

Technical or experienced users may prefer using the *MPE Quick Reference Guide* (32033-90023).

The "Commands" section of this manual lists the command syntax for all user commands, as shown in Figure 3-7. It also lists the system defaults and any special capabilities associated with that command.

```
PURGE

        Deletes a file from the system.

        PURGE filereference [,TEMP]

        purge junk.group.account
```

LG200077_035

Figure 3-7. :PURGE Listing in MPE Quick Reference Guide

In addition to listing command syntax, the *MPE Quick Reference Guide* (32033-90023) includes the following information:

- File System
- Utilities
- Intrinsics
- Segmenter
- Debug
- FCOPY
- SORT-MERGE

- EDIT
- IMAGEQUERY
- QUERY
- KSAM/ V
- V/PLUS
- Special terminal keys

The HELP Facility

The online HELP facility literally puts the *MPE V Commands Reference Manual* (32033-90006) at your fingertips. By using the HELP facility, you can find out what commands are available. You can also display information about any command, including its parameters, how it works, or an example of its use.

Using the HELP Facility

The procedures that follow outline two different ways of using the HELP facility.

Finding General Information about Commands

If you need help with a command, but you don't know the exact command or how to spell it, you can use the HELP facility to display commands in various categories. For example, if you want a command related to running a job, you could display all commands that relate to jobs.

To find a command with the HELP facility:

1. Enter HELP

   In response, MPE displays a screen similar to Example 3-1.

2. Enter one of the keywords shown at the bottom of the screen to specify the command category to be displayed.

3. Enter the desired command.

4. Enter EXIT when finished using the HELP facility.

   The colon (:) prompt reappears.

Example 3-1 shows step 1 (how to enter the HELP facility to display the available command categories).

```
:HELP

Information is available on the following classes of commands:

Running Sessions
Running Jobs
Managing Files
Running Subsystems and Programs
System Management, Status, and Accounting
Operator Control
Spooler Control
Utility Functions

For more information, enter a KEYWORD.  You can also enter any command
name as a keyword.  Enter 'help' for information on help.  Enter 'exit'
to leave help.

KEYWORDS:  SESSIONS, JOBS, PROGRAMS, FILES, MANAGE, OPERATION, SPOOLER,
           UTILITY
```

Example 3-1. Step 1: Displaying General Information

Example 3-2 shows step 2 (how to narrow the search to commands that affect files), step 3 (how to display information about a particular command) and step 4 (how to exit the HELP facility).

```
> FILES

Managing Files/Device Files.  Following are the commands used:

ALTSEC              LISTEQ              PURGEVSET           STORE
ALTVSET             LISTF               RELEASE             VSUSER
ASSOCIATE           LISTFTEMP           RENAME
BUILD               LISTVS              RESET
DISASSOCIATE        MOUNT               RESTORE
DISMOUNT            NEWVSET             SAVE
FILE                PURGE               SECURE

You can use any command as a keyword.

KEYWORDS:  SESSIONS, JOBS, PROGRAMS, FILES, MANAGE, OPERATOR, SPOOLER,
UTILITY

> PURGE

Deletes file from system

SYNTAX

     :PURGE filereference [;TEMP]

KEYWORDS:  PARMS, OPERATION, EXAMPLE

> EXIT
:
```

Example 3-2. Steps 2, 3, and 4: Displaying General Information, Displaying a Command, and Exiting the HELP Facility

Finding Specific Information about Commands

If you know the command for which you need help, you can find the following information in a single step, without having to enter and then exit the HELP facility.

- The command's syntax
- The command's parameters
- How the command works
- An example of its use

To display a command's syntax with the HELP facility:

Enter HELP, followed by the name of the command for which you want information.

```
:HELP command
```

In response, MPE displays a screen similar to Example 3-3.

```
:HELP HELLO
Initiates an interactive session

SYNTAX

   HELLO [sessionname,]username[/userpass].acctname[/acctpass]
        [groupname[/grouppass]]

          [;TERM=termtype]
          [;TIME=cpusec]
                       BS
                       CS
            [;PRI=     DS  ]
                       ES
            {;INPRI=input priority}
            {;HIPRI}

KEYWORDS:  PARMS, OPERATION, EXAMPLE
```

Example 3-3. Displaying Command Syntax

To display information about a command's parameters with the HELP facility:

Enter HELP, followed by the command name and PARMS.

```
:HELP command PARMS
```

In response, MPE displays a screen similar to Example 3-4.

```
:HELP HELLO PARMS

PARAMETERS

sessionname      Arbitrary name used in conjunction with <username> and
                 <acctname> parameters to form a session identity.  Con-
                 tains from 1 to 8 alphanumeric characters, beginning
                 with a letter.  Default is no session name is assigned.

username         A user name, established by Account Manager, that allows
                 you to log on under this account.  It contains from 1 to
                 8 alphanumeric characters, beginning with a letter.
                 (REQUIRED PARAMETER)

userpass         User password, optionally assigned by Account
                 Manager.  Contains from 1 to 8 alphanumeric
                 characters, beginning with a letter.
```

Example 3-4. Displaying Information about Parameters

To display information about a command's operation with the HELP facility:

Enter HELP, followed by the command name and OPERATION.

```
:HELP command OPERATION
```

In response, MPE displays a screen similar to Example 3-5.

```
:HELP HELLO OPERATION

Initiates an interactive session.  MPE prompts with colon when it is
ready to accept commands.  :HELLO command must be entered from a termi-
nal, no other device can be used for this command.

If passwords are omitted, MPE prompts for each one individually.  When
MPE initiates the session, it displays HP 3000, the MPE version, and the
date and time, then prompts for commands with a colon as follows:

          :HELLO ED.MPE
          HP 3000 / MPE IV C.00.00.  TUE, APR 10, 1979, 12:15 PM

KEYWORDS:  PARMS,OPERATION,EXAMPLE
:
```

Example 3-5. Displaying Information about how a Command Works

To display an example of the command with the HELP facility:

Enter HELP, followed by the command name and EXAMPLE.

```
:HELP command EXAMPLE
```

In response, MPE displays a screen similar to Example 3-6.

```
:HELP HELLO EXAMPLE

Example

To start a session named ALPHA, with the username MAC, accountname
TECHPUBS, group XGROUP, grouppass XPASS, enter:

:HELLO ALPHA/MAC.TECHPUBS,XGROUP/XPASS
```

Example 3-6. Displaying a Command Example

Overview of Common Commands

The commands listed in this section give you an idea of the scope of what you can do. They are explained in more detail in the chapters that follow.

Monitoring Commands

The monitoring commands display information that lets you monitor your job or session. These commands show information only; they do not affect files or system information. For sample displays and additional information, refer to Chapter 8, "Running Sessions."

Table 3-1 shows the most commonly used monitoring commands.

Table 3-1. The Monitoring Commands

Command	Function
SHOWALLOW	Displays Operator commands available to specific users.
SHOWTIME	Shows the day, date, and time. The Operator sets this clock, so its accuracy depends on that of the Operator.
SHOWJOB	Lists information about all jobs and sessions on the system. The parameters (if any) determine what information is displayed.
SHOWDEV	Shows the status of all devices. If followed by a peripheral's logical device number, it shows information about that particular device.
SHOWME	Displays information about the terminal's current session, including the session's user name and account name.
REPORT	Displays groups in user's account.

File Commands

The file commands allow you to create and manipulate files. Files are defined as a set of records that have a marked beginning and end. Files may be permanent or temporary. Like files in your filing drawer, files are organized by name.

In addition to the file commands, various subsystems also manipulate files. For example, you can create files with the EDITOR subsystem or copy them with the FCOPY subsystem.

File handling is discussed in more detail in Chapter 7, "Working with Files;" subsystems in Chapter 5, "Utilities and Subsystems."

Table 3-2 shows the most commonly used file commands.

Table 3-2. The File Commands

Command	Function
BUILD	Creates files and specifies their characteristics.
LISTF	Lists permanent files in user's group, in another group, or in another account, as well as information about files.
LISTFTEMP	Lists temporary files.
FILE	Creates "file equations," which tell the computer where a file is and what to do with it (usually where to print it).
LISTEQ	Lists file equations created with :FILE command.
RENAME	Allows creator of a file to give it a different name.
PURGE	Deletes files from disc.
RESET	Deletes file equations.
SAVE	Makes temporary files permanent.

Session Commands

The session commands manage the communication between your terminal and MPE in interactive sessions. Sessions are discussed in detail in Chapter 8, "Working in Sessions."

The most commonly used session commands are summarized in Table 3-3.

Table 3-3. The Session Commands

Command	Function
HELLO	Identifies you as a legitimate user and connects you to the HP 3000, in a process known as "logging on."
REDO	Corrects spelling or punctuation errors in the last command entered. Can also be used to repeat commands.
SETMSG	Turns on "quiet mode," which prevents other users from interrupting your work with the :TELL command.
ABORT	Exits from programs or subsystems after Break is pressed.
RESUME	Returns to programs or subsystems after Break is pressed.
RUN	Executes programs or subsystems.
BYE	Terminates a session and disconnects you from the HP 3000, which is known as logging off.
HELP	Gives you access to the HELP facility.
SETCATALOG	Enables and disables UDC files.
SHOWCATALOG	Displays available UDC files.

Job Commands

The job commands let you create and run batch jobs. Jobs are discussed in detail in Chapter 9, "Jobs."

The most commonly used job commands are summarized in Table 3-4.

Table 3-4. The Job Commands

Command	Function
:JOB	First command in a jobfile. Requires user and account name. Can also include other optional parameters.
!COMMENT	Lets you include explanatory comments, notes, or headings.
!TELL	Displays a message at a user terminal.
!TELLOP	Displays a message at the Operator's console.
!CONTINUE	Prevents the command that follows it from aborting the job.
!EOJ	Last command in a jobfile that terminates the job.
:STREAM	Sends the job file to the system for processing.

Message Commands

Some message commands are used by the operations staff to send messages to users. This includes the Welcome message, which is displayed when a user logs on. It is often used for important system messages, such as scheduled downtime or communications problems. The operations staff can also send Warning messages that interrupt the user's application with an important message. It is usually sent in an emergency or just before the system is brought down.

You can send messages as well, to either the Operator or to other users who are logged on.

Table 3-5. The Message Commands

Command	Function
TELLOP	Allows you to send messages to the operations staff.
TELL	Allows you to send messages to other users (unless they used the :SETMSG command to keep from being interrupted).

4

UDCs

Chapter Overview

UDCs (User Defined Commands) are shortcuts that let you rename or combine commands to suit your own needs.

This chapter includes the following topics:

- An introduction to UDCs
- The UDC levels (system, account, and user)
- UDC components
- Creating UDCs
- Creating "nested" UDCs
- Adding, changing, and deleting UDCs
- Displaying information about existing UDCs
- Using the HELP facility with UDCs

Introduction

UDCs are powerful tools that can be used in sessions, jobs, or programs. Most Account and System Managers set up UDCs that are available to all account or system users. In addition, you can set up UDCs for your personal use, to simplify complex or repetitive tasks.

For example, you can create a UDC that executes a long command with several parameters when you enter a single letter. Or you can create a UDC that executes several commands with a single command. If you often enter the same three commands in a row, for example, you could combine all three into a single command and give that command a short, descriptive name. Then, each time you enter that name, MPE executes all three commands, in the order specified. You save yourself a lot of typing (and probably a few CI error messages that point out spelling errors).

For additional information, refer to the *MPE V Commands Reference Manual* (32033-90006), as well as the *System Operation and Resource Management Reference Manual* (32033-90005).

NOTE

This chapter includes step-by-step instructions for several tasks. To avoid repetition, assume the following:

1. At each step, you enter information at the colon (:) prompt, unless specified otherwise.

2. After entering information, you press Return.

UDCs and Function Keys

UDCs are not to be confused with another MPE timesaver, the function keys. As discussed in Chapter 2, "The Terminal," you can program the terminal's function keys to enter frequently used phrases (such as the date) or single commands (such as `:LISTF`) when you press the programmed function key.

In contrast, UDCs contain sequences of commands that execute when the UDC name is entered. Also, unless you follow special procedures, function keys work as programmed only until the terminal's power is turned off. UDCs, on the other hand, are stored on disc as files. They become a part of MPE's directory until you delete them.

UDC Levels

The system has three levels of UDCs: system, account, and user. System and account level UDCs are created by the System and Account Managers, respectively. When you log on, all

system and account level UDCs are available. As a general user, you can execute these UDCs, but you cannot create, modify, or delete them.

You can only modify or delete UDCs created by a user, also called "user-level UDCs." If you think a UDC you created would benefit other users, ask your manager about making it an account or system UDC.

UDC Construction

A UDC consists of three parts; the header, the body, and the end. Figure 4-1 shows a typical UDC skeleton. Example 4-1 shows an actual UDC.

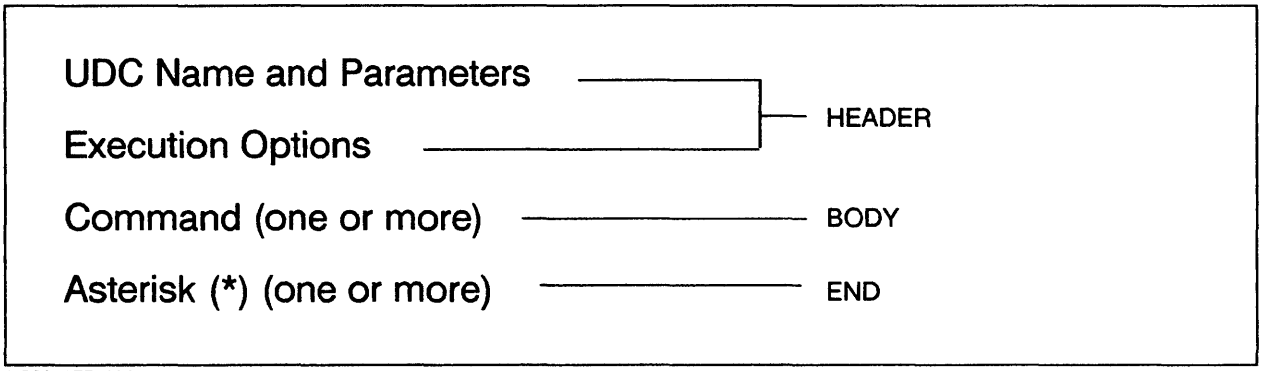


Figure 4-1. Anatomy of a UDC

Example 4-1 shows a simple UDC with the NOHELP execution option. This UDC would execute the :LISTF command, displaying all files and associated characteristics.

```
L1
OPTION  NOHELP
LISTF, 1
****
```

Example 4-1. Simple UDC Example

Header

The header consists of the UDC's name and optional parameters (if any) in the first line and execution options (if any) in the second line.

UDC Name

When creating a UDC, first assign its name. For example, you could assign the name SME to a UDC that executes the :SHOWME command. UDC names must start with a letter and can be up to 16 alphanumeric characters long. However, since the whole idea is to make things simpler, keep your names as short and descriptive as possible.

NOTE

Do not duplicate the names of existing UDCs, MPE commands, or program files.

UDC Parameters

Parameters are options that cause a general UDC to perform a specific function. You can include any parameters allowed for the command(s) included in the UDC. For example, if the UDC SME executes the :SHOWJOB command, SME16 executes the :SHOWJOB command for session 16.

Execution Options

You can specify one or more execution options, which further define how UDCs execute the commands they contain. Execution options include: LIST/NOLIST,BREAK/NOBREAK, LOGON/NOLOGON,HELP/NOHELP. If you do not specify execution options, MPE automatically assigns default values. As with most defaults, this is the most commonly used value.

LIST/NOLIST

The default is NOLIST, which means commands contained in UDCs are not displayed when the UDC executes. If you specify the LIST execution option, each command is shown on the screen as it executes.

BREAK/NOBREAK

The default is BREAK, which means all commands designated by MPE as "breakable" can be interrupted with the Break key. (For a list of "breakable" commands, see Appendix A.) If you specify the NOBREAK execution option, pressing Break has no effect.

LOGON/NOLOGON

The default is NOLOGON, which means that the UDC executes only when you enter its name, not automatically when you log on. However, if your routine includes entering the same command—or series of commands—each time you log on, specifying the LOGON execution option may save some time.

When a user logs on, MPE executes the first UDC that contains the LOGON option. Therefore, each user should have only one UDC with this option.

NOTE

If you set up a LOGON UDC, that UDC will be executed the next time you log on.

HELP/NOHELP

The default is HELP, which means the UDC's name and the commands it contains are displayed in response to the :HELP command. If you specify the NOHELP execution option, the :HELP command will not work with that UDC. As a result, other users cannot find out what commands are included in a UDC, although it may be obvious from what happens when it executes.

The NOHELP execution option is mostly used by Account or System Managers to protect system or account-level UDCs. If the NOHELP option is specified, the following response appears in response to the :HELP command:

```
CAN'T FIND ANYTHING UNDER THIS COMMAND OR IN TABLE OF CONTENTS
```

Body

The UDC's body contains the commands the UDC executes when you enter its name. It can also contain other UDCs (nested UDCs) and explanatory comment lines. To find out more about comment lines, see the *MPE V Commands Reference Manual* (32033-90006).

If a command within a UDC is more than one line long, use the ampersand (&) character immediately before or after a delimiter to continue the command into the next line.

End

The end consists of one or more asterisk (*) characters on a separate line, starting in column 1.

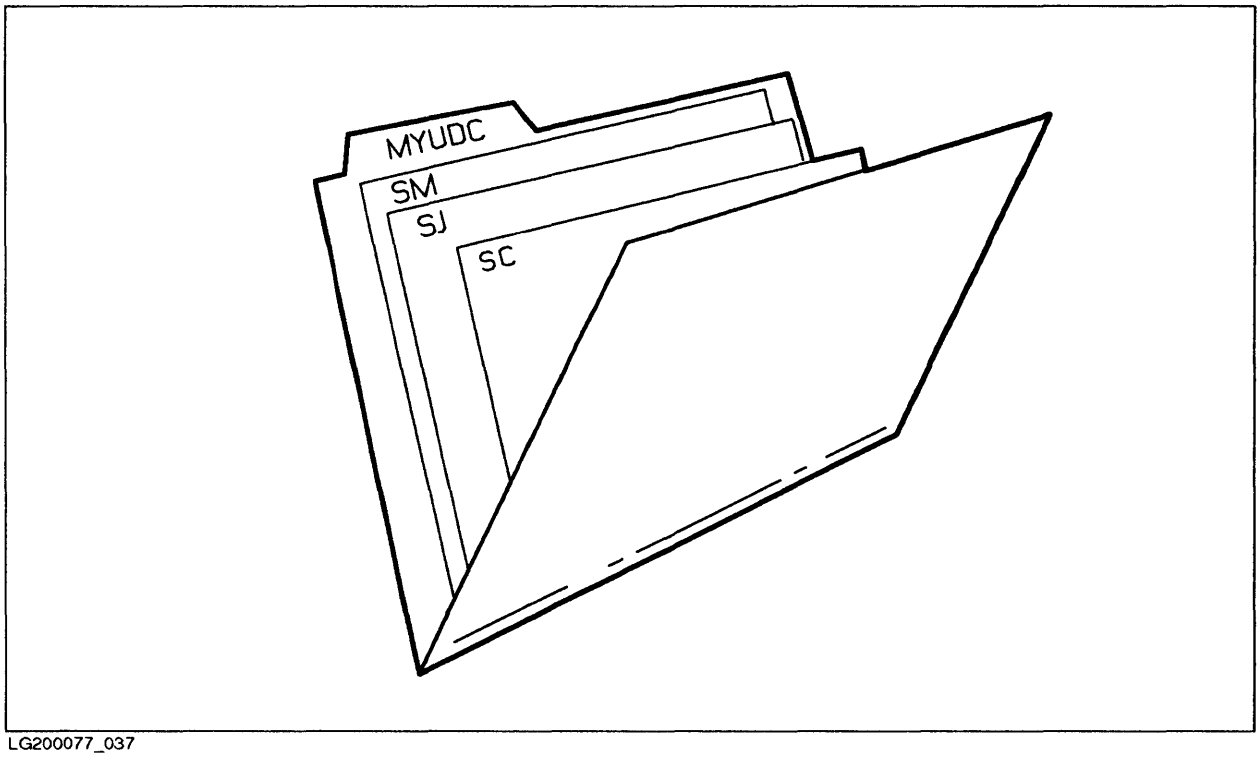
NOTE

Do not use the :REDO, :DATA, :JOB, or :HELLO commands in a UDC.

Creating UDCs

The first step in creating UDCs is to create a UDC file with EDITOR or another text processing program. Users may choose to keep all user level UDCs in a single UDC file (recommended) or in separate UDC files.

Figure 4-2 illustrates the concept of a UDC file. This file, which is named MYUDC, contains three UDCs: SM, SJ, and SC.



LG200077_037

Figure 4-2. UDC File: The Concept

Once you have created a UDC file, you must enable (catalog) the UDC file by entering the :SETCATALOG command, followed by the UDC file name. Once the UDC file is enabled, MPE recognizes and executes any UDCs in that file. However, you cannot make changes to an enabled UDC file.

NOTE

Although you could create a separate UDC file for each UDC, we strongly recommend that general users create a single UDC file that contains all UDCs. You can then add, delete, or modify UDCs as needed within that file.

This has several advantages. First, you have to remember just one file name when you want to modify the file. Second, it keeps things simple when it comes to using the :SETCATALOG command.

To create a UDC file:

1. Enter EDITOR to enter the EDITOR subsystem.
   a. Enter ADD at the slash prompt to open a file.
   b. Enter the UDC name and any parameters, if desired. As with naming files, choosing a name that makes sense helps to remember it.
   c. (Optional) Enter OPTION, followed by any execution options you want to apply to that UDC. The choices are LIST/NOLIST, HELP/NOHELP, BREAK/NOBREAK, and LOGON/NOLOGON.
   d. On a separate line, enter each command you want the UDC to execute. Include any parameters.
   e. On a separate line, enter one or more asterisk (*) characters to end that UDC.
   f. To create additional commands within this UDC file, start again at Step b.
   g. When you are finished entering UDCs, enter two slash characters (//) to close the UDC file.
   h. Enter KEEP, followed by the name of the UDC file.
   i. Enter EXIT to exit the EDITOR subsystem.
2. To enable the UDC file, enter SETCATALOG, followed by the name of the UDC file.

   The UDC file you created is now enabled; any UDCs it contains are ready to be executed.

Example 4-2 shows how to create the three UDCs shown in Figure 4-2 within a single UDC file named MYUDC. The first UDC (SM) will execute the :SHOWME command. The second UDC (SJ) will execute the :SHOWJOB command as specified by the SCHED parameter (which displays only jobs scheduled to run at a certain time). The third UDC (SC) will display the UDCs in the UDC catalog. These UDCs do not contain any execution options.

```
:EDITOR
/

/ADD

1   SM
2   SHOWME
3   ****
4   SJ
5   SHOWJOB SCHED
6   ****
7   SC
8   SHOWCATALOG
9   *****
10  ///
...

/KEEP MYUDC

/EXIT

:END OF PROGRAM

:SETCATALOG MYUDC
```

Example 4-2. Creating several UDCs within a UDC File

Creating Nested UDCs

You can also define UDCs so that entering a single UDC causes several others to execute. This process is called "nesting."

The procedure for creating nested UDCs is very similar to that for creating simple UDCs. However, the first UDC in the list contains the UDC names of all UDCs that follow. When you enter the name of this UDC, all other commands in that UDC execute as well. Since MPE V searches from the current UDC to the end of the file, any commands that invoke subsequent commands must be listed before the commands they trigger. Otherwise, the nested UDC will not execute correctly.

To create nested UDCs:

1. Enter EDITOR to enter the EDITOR subsystem.

   a. Enter ADD to open a file.

   b. Enter the UDC name that will cause the execution of other UDCs, as well as any desired parameters.

   c. (Optional) Enter OPTION, followed by any execution options you want to apply to that UDC. The choices are LIST/NOLIST, HELP/NOHELP, BREAK/NOBREAK, and LOGON/NOLOGON.

   d. On a separate line, enter the name of each UDC that you want the nested UDC to execute. Include any parameters.

   e. On a separate line, enter one or more asterisk (*) characters to end the nested UDC.

   f. Enter the name of the first UDC to be executed by the nested UDC.

   g. Enter the commands associated with that UDC.

   h. Enter one or more asterisk (*) characters to end the UDC.

   i. Repeat steps g., h., and i. until all listed UDCs are defined.

   j. When you are finished entering UDCs, enter two slash characters (//) to close the UDC file.

   k. Enter KEEP, followed by the name of the UDC file.

   l. Enter EXIT to exit the EDITOR subsystem.

2. To enable the nested UDC, enter SETCATALOG, followed by the name of the UDC file that contains the nested UDC.

Example 4-3 shows how to create a nested UDC named INFO, which causes all three UDCs (SM, SJ, and SC) to execute. Note that the nested UDCs are identical to those in the UDC file MYUDC (Example 4-2). However, to obtain the same result without nesting the UDCs, you would have to enter each separately.

```
:EDITOR
/

/ADD

1   INFO
2   SM
3   SJ
4   SC
5   ****
6   SM
7   SHOWME
8   ****
9   SJ
10  SHOWJOB SCHED
11  ****
12  SC
13  SHOWCATALOG
14  *****
15  //

...

/KEEP INFO

/EXIT

:END OF PROGRAM

:SETCATALOG INFO
```

Example 4-3. Nesting UDCs

Adding, Modifying and Removing UDCs

To add, modify, or remove user-level UDCs, you must change the contents of the UDC file. However, a UDC file that is enabled with the :SETCATALOG command cannot be changed. Therefore, you must first disable the UDC file with the :SETCATALOG command. You can then add, delete, or change any UDCs in the UDC file.

Changing the contents of a UDC file consists of three main steps:

1. Disabling the UDC file by entering the :SETCATALOG command by itself, without any file names. The UDC file becomes a "normal" file again.
2. Opening, editing, and saving the UDC file with the usual procedures.
3. Enabling the UDC file by entering the :SETCATALOG command, followed by the UDC file name.

As you can see, if you create a separate UDC file for each UDC, you have to enter each file name in Step 3 every time you want to change any one UDC file. With a dozen or so UDC files, this could become hopelessly confusing. In contrast, if all UDCs are in a single UDC file, modifying and removing UDCs is simple.

NOTE

The streamlined procedures that follow assume familiarity with the EDITOR subsystem. For further information, refer to Chapter 5, "Utilities and Subsystems."

To add a UDC to a UDC file:

1. Enter SETCATALOG to disable the UDC file.
2. Enter EDITOR to enter the EDITOR subsystem.
   a. Enter T or TEXT and the UDC file name to retrieve the UDC file from disc.
   b. Enter LIST ALL to display the file's text.
   c. Enter A or ADD
   d. Make your additions
   e. When you are finished, enter two slash characters (//).
   f. Enter K or KEEP and the UDC file name to save the file.
   g. Enter Y or YES in response to the PURGE OLD? prompt.
   h. Enter E or EXIT to exit the EDITOR subsystem.
3. Enter SETCATALOG and the UDC file name to enable the UDC file.

To modify a UDC in a UDC file:

1. Enter SETCATALOG to disable the UDC file.
2. Enter EDITOR at the colon prompt to enter the EDITOR subsystem.
   a. Enter T or TEXT and the UDC file name to retrieve the file from disc.
   b. Enter LIST ALL to display the file's text.
   c. Enter M or MODIFY, followed by the line number(s) you want to change.
   d. Make your corrections.
   e. Enter K or KEEP and the UDC file name to save the file.
   f. Enter Y or YES in response to the PURGE OLD? prompt.
   g. Enter E or EXIT to exit the EDITOR subsystem.
3. Enter SETCATALOG and the UDC file name to enable the UDC file.

Example 4-4 shows how to change the format of the :SHOWJOB display by replacing the parameter SCHED with the parameter STATUS in the UDC SJ in the UDC file MYUDC (created in Example 4-2).

```
:SETCATALOG
:

:EDITOR
/
/TEXT MYUDC

/LIST ALL

1       SM
2       SHOWME
3       ****
4       SJ
5       SHOWJOB SCHED
6       ****
7       SC
8       SHOWCATALOG
9       *****

/MODIFY 5

SHOWJOB SCHED
        RSTATUS
SHOWJOB STATUS

/LIST ALL

1       SM
2       SHOWME
3       ****
4       SJ
5       SHOWJOB STATUS
6       ****
7       SC
8       SHOWCATALOG
9       *****

/KEEP MYUDC

MYUDC ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW.
PURGE OLD?  YES

/E

:END OF PROGRAM

:SETCATALOG MYUDC
```

Example 4-4. Modifying a UDC

To delete a UDC from a UDC file:

1. Enter SETCATALOG to disable the UDC file.
2. Enter EDITOR to enter the EDITOR subsystem.
   a. Enter T or TEXT and the UDC file name to retrieve the file from disc.
   b. Enter LIST ALL to display the file's text.
   c. Enter D or DELETE, followed by the line numbers you want to delete.
   d. Enter K or KEEP and the UDC file name to save the file.
   e. Enter Y or YES in response to the PURGE OLD? prompt.
3. Enter SETCATALOG and the UDC file name to enable the UDC file.

Example 4-5 shows how to delete the UDC SJ from the UDC file MYUDC.

```
:SETCATALOG
:

:EDITOR
/
/TEXT MYUDC

/LIST ALL

1       SM
2       SHOWME
3       ****
4       SJ
5       SHOWJOB STATUS
6       ****
7       SC
8       SHOWCATALOG
9       ****

/DELETE 4/6

/LIST ALL

1       SM
2       SHOWME
3       ****
7       SC
8       SHOWCATALOG
9       ****

/KEEP MYUDC

MYUDC ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW.
PURGE OLD? YES

/E

:END OF PROGRAM

:SETCATALOG MYUDC
```

Example 4-5. Deleting a UDC from the UDC File

Finding out about UDCs on the System

The :SHOWCATALOG command shows which system-level, account-level, and user-level UDCs are available. To find out which command each UDC actually executes, you can use the HELP facility. However, if the creator of a UDC specified the NOHELP option when creating the UDC, you cannot find out which command(s) it contains.

Using the :SHOWCATALOG Command

In response to the :SHOWCATALOG command, MPE displays all user-level, account-level, and system level UDC files, as well as the UDCs they contain. This can be useful when you move to a different account and want to find out what account-level and system-level UDCs are available.

To display a list of all UDC files and the UDCs they contain:

Enter :SHOWCATALOG

In response, MPE displays a screen similar to Example 4-6. As you can see, user-level UDCs appear at the top.

Example 4-6 displays the user-level UDC file created in Example 4-2, followed by account-level and system-level UDC files and the UDCs they contain. (On an actual system, there may be so many UDCs that they take up several pages. Remember, you can stop and start the scrolling with CTRL S and CTRL Q, or the Stop key). You can also get a printout of this listing, as explained in "Printing What is on the Screen", in Chapter 2.

```
:SHOWCATALOG

MYUDC.PUB.PRXL
    SM                  USER
    SJ                  USER
    SC                  USER
CMLUDC.PUB.3000
    LOG                 ACCOUNT
    TDP                 ACCOUNT
    CMLSETUP            ACCOUNT
SYS0345.SYSTEM.SYS
    LOGON               SYSTEM
    CHAPCHECK           SYSTEM
    SLATE               SYSTEM
```

Example 4-6. Displaying UDCs with the :SHOWCATALOG Command

Using the HELP Facility with UDCs

The :SHOWCATALOG command displays available UDCs. However, it does not show what commands they execute, what execution options they contain, or whether any parameters apply. That is where the HELP facility comes in.

To find out what each UDC does:

Enter HELP and the UDC name.

```
:HELP UDC name
```

In response, MPE displays the entire UDC for which you requested help.

Example 4-7 shows how to find out what commands the system UDC SLATE contains.

```
:HELP SLATE

USER DEFINED COMMAND:

SLATE
OPTION LIST
RUN HPSLATE.PUB.SYS
```

Example 4-7. Using the HELP Facility to Display UDC Contents

NOTE

If you get the following response, the UDC's creator probably specified the NOHELP execution option.

```
CAN'T FIND ANYTHING UNDER THIS COMMAND OR IN TABLE
OF CONTENTS
```

5

Utilities and Subsystems

Chapter Overview

The HP 3000 includes a number of programs designed for different tasks, such as creating ASCII text files (text processing), backing up data, and sorting or merging files. These programs are called "utilities" or "subsystems." This chapter introduces you to those subsystems of interest to the general user, including:

- EDITOR
- STORE/RESTORE
- SORT/MERGE
- FCOPY
- SPOOK

NOTE

The DSCOPY utility, which is part of the DS/3000 Data Communication subsystem, lets you transfer files between different HP 3000 systems. Although this utility must be purchased separately, Appendix B contains procedures for those users who need them.

Introduction

The terms "utility" and "subsystem" are often used interchangeably. Both are programs within MPE that perform specific tasks. In general, utilities are more specific and less complex. Utilities are frequently used by programmers and the operations staff to maintain the system. Subsystems, as their name implies, often have prompts and commands of their own. For simplicity, this manual calls them all "subsystems."

Some subsystems are not used by general users. Others, like the EDITOR subsystem, are used by many different users. This chapter describes the most commonly used subsystems.

NOTE

This chapter includes step-by-step instructions for several tasks. To avoid repetition, assume the following:

1. At each step, you enter information at the colon (:) prompt, unless specified otherwise.
2. After typing the information, you press Return.

Table 5-1 summarizes the functions of the subsystems covered in this chapter, as well as the prompts that identify them.

Table 5-1. Subsystem Prompts and Functions

Subsystem	Prompt	Function
EDITOR	/	Text processing.
SORT	>	Sorts files according to selected criteria.
MERGE	>	Merges any sorted files.
STORE	>	Copies files to tape for backup and transfer.
RESTORE	>	Copies files copied to tape with STORE back to disc.
FCOPY	>	Copies files between groups and accounts on same system, copies files from one medium to another, compares file contents, and more.
SPOOK	>	Allows access to and manipulation of spoolfiles.

NOTE

To use most of these subsystems, enter `:RUN`, followed by the subsystem's name. For example, to use the SORT subsystem, you would enter `:RUN SORT.PUB.SYS`.

However, many System Managers find it helpful to develop shortcuts (UDCs) that require only the subsystem's name, or even an abbreviation, to run it. If such a UDC is set up in your system, you could run the subsystem by entering `SORT` or even `SO`.

EDITOR

The EDITOR subsystem is described in detail in the *EDIT/3000 Reference Manual* (03000-90012). Because the EDITOR text processing subsystem is included with every HP 3000, this manual provides basic instructions for its use. However, after the system is set up, you can use many other word processors on the HP 3000.

The EDITOR subsystem lets you create, edit, and save files. These files may be documents like memos, letters, or manuals. They can also be programs or spreadsheets. In many ways, EDITOR works like a typewriter, except that the words appear on the screen instead of paper.

Using EDITOR involves four basic steps, which are discussed later in more detail:

- Entering the EDITOR subsystem.
- Creating a new file or modifying an existing one.
- Saving the file with the /KEEP command.
- Leaving the EDITOR subsystem.

You will find procedures for renaming and deleting files in Chapter 7, "Working with Files." You will use MPE commands, rather than EDITOR subcommands, for these tasks.

NOTE

You can execute most MPE commands, including :LISTF, :SHOWJOB, :SHOWME, and :SHOWOUT without leaving the EDITOR subsystem. Simply enter the colon (:) prompt after EDITOR's slash (/) prompt, followed by the desired MPE command.

About Line Numbers

EDITOR is a "line editor" text processing program. That means that the program numbers each line as you add text. To edit the file, use these line numbers to tell the computer which lines to edit. Whatever EDITOR command you use affects *only* the line number(s) specified. For example, you can add a word to line 4 by specifying that line. In the same way, you could specify a range of text by asking for lines 10 through 20.

Specifying Line Numbers

The procedures that follow give specific examples of how to specify line numbers with various commands. In general, line numbers are specified using the following notation:

- n a single line number.

- n, m two different line numbers.
- n/m a range of line numbers, including the first and last numbers.
- ALL all line numbers within the file.

Saving Line Numbers

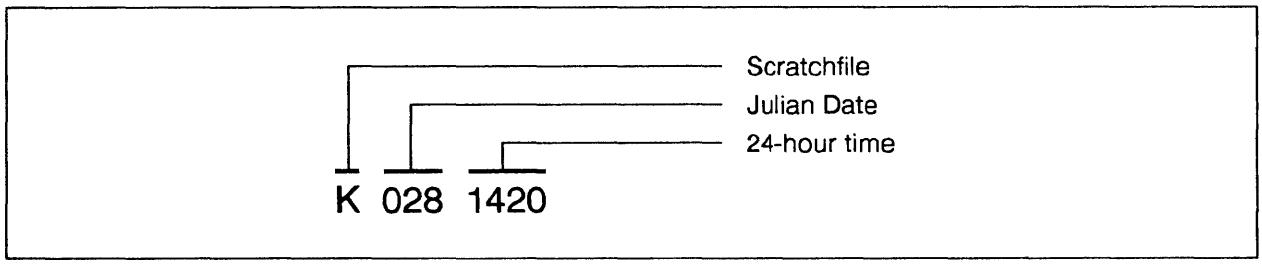
The /KEEP command lets you save files created in EDITOR. Unless you specify otherwise, EDITOR stores the information entered with the line numbers. However, certain tasks will not work unless you save the file without line numbers. For example, the SORT program requires "unnumbered" EDITOR files.

You can specify whether or not to keep line numbers in the /KEEP command by adding the "unnumbered" parameter. (Line numbers will still appear at the beginning of each line to make editing possible).

About Scratch Files

The EDITOR subsystem creates temporary files, called "scratch" files or "K" files when the program is unable to close the file properly for any reason. For example, EDITOR would save the current file as a scratch file in case of a system failure. Before logging off, you could then save the file properly with the /KEEP command.

You may see such scratch files in your account when you execute the :LISTF command. These files are preceded by the letter K and followed by seven numbers, which identify the date and time they were created. The first three numbers are the Julian date (days are numbered 1 through 365); the second four numbers list the time in 24-hour notation. For example, the scratch file shown in Figure 5.1 was created on January 28, at 2:20 p.m.



LG200077_038

Figure 5-1. Anatomy of a Scratch File

Using EDITOR

As with MPE, you use commands to communicate with the EDITOR subsystem. EDITOR includes 29 commands, which are explained in detail in the *EDIT/3000 Reference Manual* (03000-90012). This section summarizes the most commonly used commands and procedures. Of course, since doing a task clarifies anything you may read about it, we encourage you try these procedures on your terminal.

Table 5-2. Summary of EDITOR Commands

Command	Shortcut	Function
ADD	A	Allows text entry by adding lines at bottom of file.
COPY	C	Copies lines from one section of file to another.
DELETE	D	Deletes the line number(s) specified.
EDITOR		Starts the EDITOR subsystem.
EXIT	E	Exits the EDITOR Subsystem.
//		Exits text entry (add) mode.
GATHER	G	Renumbers lines or moves lines within file.
KEEP	K	Saves file to disc. With the parameter *unn* (for "unnumbered,") the file is saved without line numbers.
LIST	L	Displays all or selected lines in file.
MODIFY	M	Allows modification of specified lines, using the subcommands D, I, and R.
	D	Deletes the letter above the D.
	I	Inserts whatever follows the I behind the letter above it.
	R	Replaces the letter above it with whatever follows the R.
TEXT	T	Retrieves file from disc and copies it into scratch file.
XPLAIN	X	Shows syntax and usage for EDITOR commands.

Creating a New File

EDITOR lets you create files--the basic building blocks for working with the computer. These files can be documents, programs, databases, and so on. Once created, you can manipulate these files with other subsystems or MPE commands. Among other things, you can edit, sort, print, and delete them.

To create a new file:

1. Enter EDITOR to enter the EDITOR subsystem.

   The slash (/) prompt appears, indicating that you are now working with EDITOR.

2. Enter ADD or A to enter add mode, which allows you to add lines.

   A numbered line appears.

3. Enter text until you get to the end of the line. Then press Return. Continue entering text until you are finished.

   New numbered lines appear each time you press Return.

4. Enter two slashes to exit add mode.

   The system displays three dots and the slash prompt.

5. Enter KEEP or K, followed by a file name, to save the document.

6. Enter EXIT or E to leave the EDITOR subsystem.

   The colon prompt reappears.

Example 5-1 illustrates using Steps 1 though 6 to create a file called SAMPLE.

```
:EDITOR
/

/ADD

1   Let's get started.
2   Come on, come on.
3   Are we having fun yet?
4   Can we take a break?
5   Enough already.
6   OK, then, let's quit.
7   //

...

/KEEP SAMPLE

/EXIT

END OF PROGRAM
```

Example 5-1. Creating a New File with EDITOR

NOTE

Although the arrow keys appear to move the cursor in the EDITOR subsystem, the system does not recognize this movement. To move the cursor in EDITOR, use the Back space and Space Bar.

Modifying an Existing File

One of the advantages of computers is that they simplify making changes. Instead of repeatedly re-entering documents, you simply correct them on the screen before printing.

To modify an existing file:

1. Enter EDITOR to enter the EDITOR subsystem.

   The slash (/) prompt appears.

2. Enter TEXT or T, followed by the file name, to retrieve the file from disc.

3. Enter LIST or L, followed by the line number(s) you want to display.

   For example,

   - LIST ALL displays all lines in the file.
   - LIST 3 displays line 3 .
   - LIST 3/9 displays lines 3 through 9 .

4. Enter MODIFY, followed by the line number you want to edit.

   You are now in Modify mode, which allows you to edit your file, one line at a time, with the DELETE (D), INSERT (I), and REPLACE (R) commands. To use these commands, follow the instructions below:

   a. To delete selected characters, move the cursor under the character(s) you want to delete and enter D . You can also enter D under the first and last letters you want to delete. All letters in between are deleted as well.

   b. To insert text, move the cursor under the character where you want to insert and enter I, followed by the character(s) you want to insert.

   c. To replace text, move the cursor under the first character(s) you want to replace and enter R, followed by the characters you want as replacements.

   d. To see the changes, press Return .

5. Press Return again to display a new slash prompt.

6. Enter KEEP or K, followed by the file name, to save the changes you made.

   The system displays: file name ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW. PURGE OLD?

7. Enter YES or Y to save the modifications. (If you enter NO or press Return , the modifications are not saved.)

8. Enter EXIT or E to leave the EDITOR subsystem.

Example 5-2 shows how to modify the file created in Example 5-1. Specifically, it shows how to delete the comma and the words "come on" from line 2, and how to add "We said:" to line 5. (Note that spaces are designated by ^ characters.)

```
:EDITOR
/
/TEXT SAMPLE

/LIST ALL

   1   Let's get started.
   2   Come on, come on.
   3   Are we having fun yet?
   4   Can we take a break?
   5   Enough already.
   6   OK, then, let's quit.

/MODIFY 2

   Come on, come on.
           D     D
   Come on

/MODIFY 5

   Enough already
   IWe said:^^
   We said:  Enough already.

/LIST ALL

   1   Let's get started.
   2   Come on
   3   Are we having fun yet?
   4   Can we take a break?
   5   We said:  Enough already.
   6   OK, then, let's quit.


/KEEP SAMPLE

SAMPLE ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW.
PURGE OLD? YES

/EXIT
```

Example 5-2. Modifying an Existing File with EDITOR

Inserting Text into Existing Files

Adding text to an existing file is similar to creating a new file. When you type A to enter add mode, EDITOR displays the next available line number at the end of the file. For example, if the last line number in the existing document is 114, line number 115 would appear, ready for you to enter text.

However, inserting text within the file is slightly more complicated. You can insert text by specifying line increments to "squeeze" up to 999 lines of text between any two existing lines. For example, specifying "1.1" lets you add nine additional lines (1.1 through 1.9) between lines 1 and 2. Specifying "1.01" lets you enter up to 99 additional lines (1.01 through 1.99), and so forth.

To insert text between lines:

1. Enter EDITOR to enter the EDITOR subsystem.
2. Enter TEXT or T at the slash (/) prompt, followed by the file's name.
3. Enter ADD or A to enter add mode. Then specify the line number where you want to insert text and the desired increment.

   A new, incrementally numbered line appears. For example, if you specified line "6.1," a new line numbered 6.1 appears in the document.
4. Enter text until you get to the end of each line and press Return. Continue entering text until you are finished.

   New, incrementally numbered lines (6.1, 6.2, 6.3, etc.) appear each time you press Return.
5. Enter two slashes (//) to exit add mode.

   The system displays three dots and the slash prompt.
6. (Optional) Enter GATHER ALL to renumber the document.

   The system renumbers the document (line 6.1 becomes line 7, line 6.2 becomes line 8, etc.)
7. (Optional) Enter LIST ALL to check your work.

   The system displays the modified, renumbered file.
8. Enter KEEP or K, followed by the file name, to save the document.

   The system displays: file name ALREADY EXISTS. RESPOND YES TO PURGE OLD AND KEEP NEW. PURGE OLD?
9. Enter YES or Y to save the modifications.

   The system replaces the old file with the updated file that includes the inserted lines.
10. Enter EXIT or E to leave the EDITOR subsystem.

Example 5-3 illustrates how to insert two lines after line 1 and renumber the lines in the SAMPLE file created in Example 5-1 and modified in Example 5-2.

```
:EDITOR
/
/TEXT SAMPLE

/ADD 1.1

   1.1   We don't have much time and
   1.2   there's so much to do.
   1.3   //

...

/GATHER ALL

/LIST ALL

   1    Let's get started.
   2    We don't have much time and
   3    there's so much to do.
   4    Come on.
   5    Are we having fun yet?
   6    Can we take a break?
   7    We said:  Enough already.
   8    OK, then, let's quit.

/KEEP SAMPLE

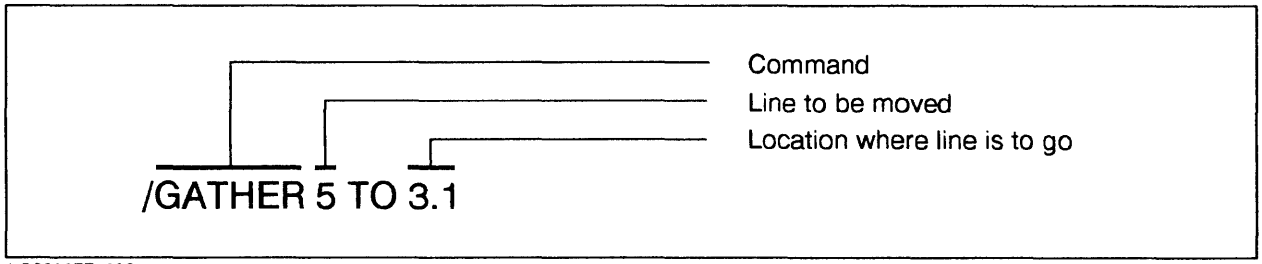
SAMPLE ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW.
PURGE OLD? YES

/EXIT
:
```

Example 5-3. Inserting Text and Renumbering Lines in an Existing File

Moving Lines within the File

You can also rearrange lines within a file with the /GATHER command. This command lets you specify the line you want to move and the location to which you want to move it, as shown below. The line no longer exists in the original position.



LG200077_039

Figure 5-2. Using the /GATHER Command

To move lines within text:

1. Enter EDITOR to enter the EDITOR subsystem.
2. Enter TEXT or T, followed by the file's name, to retrieve the file.
3. Enter LIST or L, followed by the line numbers you want to display.
4. Enter GATHER, followed by the line number(s) to be moved and their destination.

   For example,

   - GATHER 6 to 10.1 moves line 6 after line 10
   - GATHER 12/15 to 8.1 moves lines 12, 13, 14, and 15 after line 8.
5. Repeat Step 4 until you have moved all lines.
6. (Optional) Enter GATHER ALL to renumber the lines in sequence.
7. (Optional) Enter LIST ALL to inspect the results.
8. Enter KEEP or K, followed by the file name, to save the document.

   The system displays: file name ALREADY EXISTS. RESPOND YES TO PURGE OLD AND KEEP NEW. PURGE OLD?
9. Enter YES or Y to confirm the changes.
10. Enter EXIT or E to leave the EDITOR subsystem.

Example 5-4 illustrates how to move line 4 after line 1 in the SAMPLE file. The /LIST command displays the result.

```
/LIST 1/4

   1    Let's get started.
   2    We don't have much time and
   3    there's so much to do.
   4    Come on.

/GATHER 4 TO 1.1
     4   =>   1.1

/LIST 1/5

   1    Let's get started.
   1.1  Come on.
   2    We don't have much time and
   3    there's so much to do.
   5    Are we having fun yet?

/KEEP SAMPLE

SAMPLE ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW.
PURGE OLD? YES

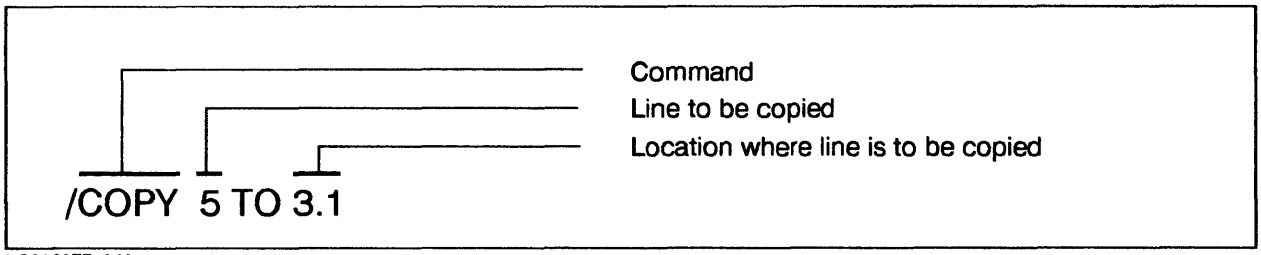
/EXIT
```

Example 5-4. Moving Lines Within a File

Copying Lines within the File

You can copy parts of the file to any other part of the file with the /COPY command. This can include single lines or groups of lines. In contrast to the /GATHER command, the line still exists in the original position, as well as the new position.

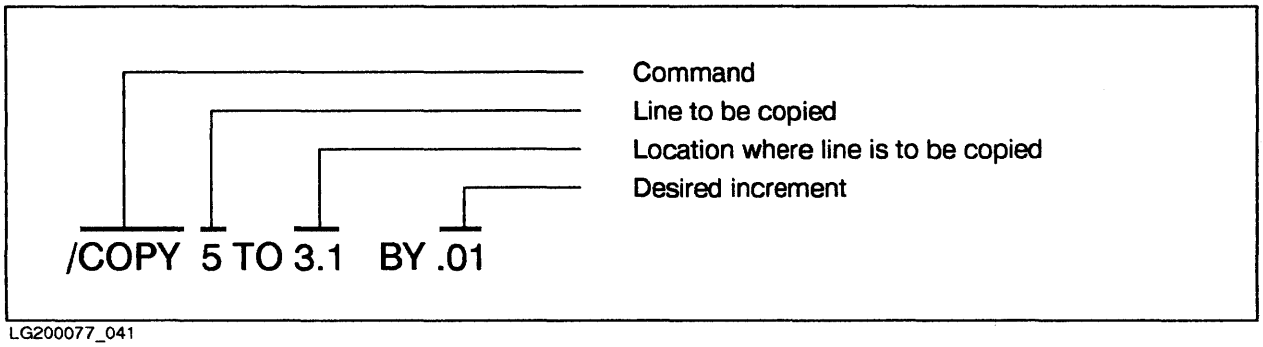
To copy fewer than 10 lines, use the command as shown below.



LG200077_040

Figure 5-3. Using the /COPY Command (1)

To copy more than ten lines, specify a smaller increment, as shown below.



LG200077_041

Figure 5-4. Using the /COPY Command (2)

To copy lines within the file:

1. Enter EDITOR to enter the EDITOR subsystem.
2. Enter TEXT or T, followed by the file's name, to retrieve the file.
3. Enter LIST or L, followed by the line numbers you want to display.
4. Enter COPY, the line number(s) to be copied, and their destination.

   For example,

   - COPY 7 TO 22.1 moves line 7 after line 22.
   - COPY 10/16 TO 50.1 moves lines 10 through 16 after line 50.
   - COPY 10/50 TO 100.01 BY.01 moves lines 10 through 50 after line 100.
5. Repeat Step 4 until you have copied all lines.
6. (Optional) Enter GATHER ALL to renumber the lines in sequence.
7. (Optional) Enter LIST ALL to inspect the results.
8. Enter KEEP or K, followed by the file name, to save the document.

   The system displays: file name ALREADY EXISTS. RESPOND YES TO PURGE OLD AND KEEP NEW. PURGE OLD?
9. Enter YES or Y to confirm the changes.
10. Enter EXIT or E to leave the EDITOR subsystem.

Example 5-5 shows how to copy line 2 after line 4 in the SAMPLE file. The /LIST command shows the result.

```
/LIST ALL

   1    Let's get started.
   2    Come on.
   3    We don't have much time and
   4    there's so much to do.
   5    Are we having fun yet?

/COPY 2 TO 4.1
     2   =>   4.1

/LIST 1/5

   1    Let's get started.
   2    Come on.
   3    We don't have much time and
   4    there's so much to do.
   4.1  Come on.
   5    Are we having fun yet?

/KEEP SAMPLE

SAMPLE ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW.
PURGE OLD? YES

/EXIT
```

Example 5-5. Copying Lines Within a File

Deleting Lines within a File

You can delete selected lines with the /DELETE command. This command is particularly useful for deleting all lines in a file to start over.

NOTE

To delete an entire file, do not use the EDITOR command /DELETE. Although you can delete a file's contents, the file still exists on disc.

To delete files, use the MPE command :PURGE, as described in Chapter 7, "Working with Files".

To delete lines from a file:

1. Enter EDITOR to enter the EDITOR subsystem.
2. Enter TEXT or T, followed by the file's name, to retrieve the file.
3. (Optional) Enter LIST or L, followed by the line number(s) you want to display.
4. Enter DELETE or D, followed by the line number(s) you want to delete.

   For example,

   - DELETE 7 removes line 7.
   - DELETE 2,5 removes lines 2 and 5
   - DELETE 10/35 removes lines 10 through 35.
   - DELETE ALL removes all lines so you can start over.
5. (Optional) Enter GATHER ALL to renumber the lines in sequence.
6. (Optional) Enter LIST ALL to see the results.
7. Enter KEEP or K, followed by the file name, to save the document.

   The system displays: `file name ALREADY EXISTS. RESPOND YES TO PURGE OLD AND KEEP NEW. PURGE OLD?`
8. Enter YES or Y to confirm the changes.
9. Enter EXIT or E to leave the EDITOR subsystem.

Example 5-6 shows how to delete all lines except the first from the SAMPLE file.

```
:EDITOR
/
/TEXT SAMPLE

/LIST 1/8

   1    Let's get started.
   2    Come on.
   3    We don't have much time and
   4    there's so much to do.
   4.1  Come on.
   5    Are we having fun yet?
   6    Can we take a break?
   7    Enough already.
   8    OK, then, let's quit.

/DELETE 2/8

   2    Come on.
   3    We don't have much time and
   4    there's so much to do.
   4.1  Come on.
   5    Are we having fun yet?
   6    Can we take a break?
   7    Enough already.
   8    OK, then, let's quit.

/LIST ALL

   1     Let's get started.

/KEEP SAMPLE

SAMPLE ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW.
PURGE OLD? YES

/EXIT
```

Example 5-6. Deleting Lines Within a File

NOTE

If you forget to save your document with the /KEEP command, the system displays the following message when you try to leave the EDITOR subsystem:

```
IS IT OK TO CLEAR ?  RESPOND "YES".  CLEAR?
```

To save your changes, enter Y or YES. To lose the changes and save the file in its original form, enter N or NO.

Printing Files

To MPE, printing simply means displaying a file on the printer instead of the terminal screen. You can print a file by adding the parameter "OFFLINE" to the /LIST command. You can also print the file on a different printer by writing a file equation, as described in Chapter 7, "Working with Files".

To print a file:

1. Enter EDITOR to enter the EDITOR subsystem.
2. Enter TEXT or T, followed by the file's name, to retrieve the file.
3. Enter LIST ALL, offline to print the file.

The terminal displays OFFLINE LISTING BEGUN

Example 5-7 shows how to print the SAMPLE file.

```
:EDITOR
/
/TEXT SAMPLE
/LIST ALL, OFFLINE
***OFF LINE LISTING BEGUN. ***
```

Example 5-7. Printing a File

NOTE

You can also use the FCOPY or DSCOPY subsystems to print files, without going into the EDITOR subsystem. For procedures, refer to the "FCOPY" section or to Appendix B.

Using EDITOR's HELP Facility

EDITOR has its own online HELP facility. Like the MPE HELP facility, it displays the proper command syntax and gives an example of the command's use.

To use EDITOR's HELP facility:

Enter X or XPLAIN, followed by the command name for which you want help.

Example 5-8 shows how to display information about the /MODIFY command.

```
/X MODIFY

    TO MODIFY TEXT IN THE WORK FILE BY USING THREE OPERATIONS OF
    DELETE (D), INSERT (I), OR REPLACE (R).
    EXAMPLE: MODIFY 50/100
```

Example 5-8. Using EDITOR's HELP Facility

SORT-MERGE

The SORT-MERGE program is described in detail in the *SORT-MERGE/3000 Reference Manual* (32214-90002).

The SORT-MERGE subsystem consists of two programs: SORT and MERGE. The SORT programs sorts information in a file according to selected criteria. You can then use the MERGE program to combine several files that contain sorted information into a single file.

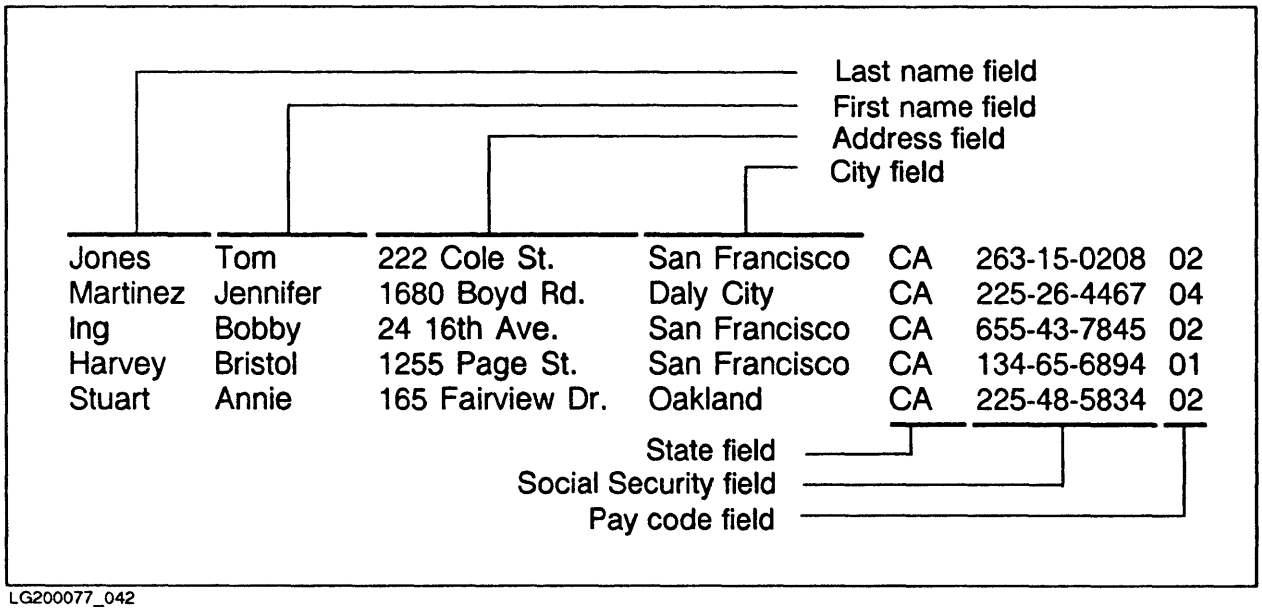
A common use for the SORT-MERGE subsystem includes creating and structuring records within files, such as employee records. For example, a company might have a record for each employee, which might include the employee's name, address, social security number, number of tax exemptions, pay scale, type of insurance, and so on. Once this information is assembled in a consistent format for each employee, the SORT program can rearrange this information in many different ways. For example, employees could be listed alphabetically, by the type of insurance coverage, or by how much money they make, starting from the lowest salary to the highest, or vice versa.

In a large company, each department might have its own file that contains records for its employees. By using the MERGE program, you could combine these sorted files into a single file for all employees within the company.

Required File Format

Each file contains a number of "records," which in turn consist of various "fields." In an employee file, for example, the information for each employee would be a record. Each piece of information within the record, such as the last name, would be a field. The records that make up files to be sorted must be in a consistent format.

Example 5-5 shows a sample file that consists of simplified employee records. The numbers at the bottom show the position for each character.



LG200077_042

Figure 5-5. Employee File with Records and Fields

Creating Files to be Sorted and Merged

When you create a file to be sorted (and perhaps merged), you must first determine the "record format," which must be the same of all records. To do this, make a list of each piece of information you want to include and the starting column, the ending column, and the total character length. The record format for Example 5-5 looks like this:

Last name	(1 - 10 = 10 characters)
First name	(11 - 20 = 10 characters)
Address	(21 - 34 = 14 characters)
City	(35 - 47 = 13 characters)
State	(48 - 51 = 4 characters)
Social Security number	(52 - 63 = 12 characters)
Pay code	(64 - 65 = 2 characters)

When determining the record format, be sure to provide sufficient space for the largest pieces of information in your records. For example, the field that will contain addresses should be large enough for the longest address among the records, as well as any addresses you might want to add in the future.

You can use the EDITOR subsystem to create the files to be sorted. However, since it is essential that each field within each record start at the same position, the procedures that follow also include instructions for setting tabs. Using tabs makes the entire process much easier.

To create a file to be sorted:

1. Enter EDITOR to enter the EDITOR subsystem.
2. Set tabs for the beginning position of each field. (We recommend using the % character as the tab character.)

   ```
   /SET TABCHAR="%", TABS=(position, position, ...)
   ```

   For example, the positions for the record format shown in Example 5-9 would be 11, 21, 35, 48, 52, 64.
3. Enter VERIFY TABS, TABCHAR to check the tab settings.

   The system displays:

   ```
   TABS = (position, position, ..)
   TAB CHARACTER = "%"
   ```
4. Enter ADD or A to enter Add mode.

   A numbered line appears.
5. Enter information for each field in the record. After each field, enter the "tab character" set up in Step 2.

   For example, the first record shown in Example 5-5 would look like this:

   ```
   1  Jones%Tom%222 Cole St.%San Francisco%CA%263-15-0208%02
   ```
6. Enter two slashes (//) to exit add mode.

   The system displays three dots and the slash prompt.
7. (Optional) Enter LIST ALL to display the file.
8. Enter KEEP or K, followed by the file name and the parameter "unn" (for "unnumbered").
9. Enter EXIT or E to leave the EDITOR subsystem.

Example 5-9 illustrates how to create a variation of an employee file. This file, named SFLIST, lists the last name, first name, address, social security number, and pay code of several employees in the San Francisco branch of a company.

```
:EDITOR
/

/SET TABCHAR="%", TABS=(11, 21, 35, 48, 52, 64)

/VERIFY TABS, TABCHAR
       TABS=(11, 21, 35, 48, 52, 64)
       TAB CHARACTER= "%"

/ADD

1  Jones%Tom%222 Cole St.%SanFrancisco%CA%263-15-0208%02
2  Martinez%Jennifer%1680 Boyd Rd.%Daly City%CA%225-26-4467%04
3  Ing%Bobby%24 16th Ave.%San Francisco%CA%655-43-7845%02
4  Stuart%Annie%165 Fairview Dr.%San Francisco%CA%225-48-5834%02
5  Harvey%Bristol%1255 Page St.%Oakland%CA%134-65-6894%01
6  //

...

/LIST ALL

Jones      Tom        222 Cole St.       San Francisco    CA   263-15-0208   02
Martinez   Jennifer   1680 Boyd Rd.      Daly City        CA   225-26-4467   04
Ing        Bobby      24-16th Ave.       San Francisco    CA   655-43-7845   02
Stuart     Annie      165 Fairview Dr.   San Francisco    CA   225-48-5834   02
Harvey     Bristol    1255 Page St.      Oakland          CA   134-65-6894   01

/KEEP SFLIST, unn

/EXIT
```

Example 5-9. Creating a File to be Sorted

Using the SORT Program to Sort Files

Now that you have an unnumbered file that consists of records and fields, you can sort it. Unless you specify otherwise, fields that contain letters are sorted alphabetically from A to Z; fields with numbers in ascending order (smallest to largest). You can also sort files for more than one field. For instructions, refer to *SORT-MERGE/3000 Reference Manual* (32214-90002).

To sort a file, specify the file to be sorted (input file), assign a name to the sorted output file, and define the field by which the file is sorted, including its starting position and the length of the field. For example, to sort the file in Example 5-9 by last name, you would specify SFLIST as the input file, assign a name to the output file, and tell MPE to sort the field that starts at position 1 and is 10 characters long. This procedure is illustrated in Example 5-10.

To sort files:

1. Enter `RUN SORT.PUB.SYS` to start the subsystem.

   The system displays the chevron (>) prompt.

2. Enter `INPUT`, followed by the name of the file to be sorted (input file).

3. Enter `OUTPUT`, followed by the name of the file that will contain the sorted material (output file).

4. Enter `KEY` and specify the starting position of the field to be sorted, followed by the number of characters in that field.

   For example, to sort by the first field in Example 5-10, enter `KEY 1, 10.`

5. Enter `END` to start the SORT operation.

   The system displays a number of statistics and the colon (:) prompt. This indicates that the SORT operation is completed.

6. (Optional) Enter the EDITOR subsystem and display the sorted file.

Example 5-10 shows how to sort, by last name, the file created in Example 5-19. The file to be sorted is called SFLIST, the sorted file SFFINAL. Example 5-10 also shows how to display the sorted file with the EDITOR subsystem.

```
:RUN SORT.PUB.SYS
>

>INPUT SFLIST

>OUTPUT SFFINAL

>KEY 1, 10

>END
:

:EDITOR
/

/T SFFINAL

/LIST ALL

Harvey    Bristol    1255 Page St.      San Francisco    CA  134-65-6894   02
Ing       Bobby      24 16th Ave.       Daly City        CA  655-43-7845   01
Jones     Tom        222 Cole St.       San Francisco    CA  263-15-0208   02
Martinez  Jennifer   1680 Boyd Rd.      San Francisco    CA  225-26-4467   04
Stuart    Annie      165 Fairview Dr.   Oakland          CA  225-48-5834   02

/EXIT
:
```

Example 5-10. Sorting a File and Displaying the Sorted File

Displaying or Editing a Sorted File

To check the sorted file, use the EDITOR subsystem to retrieve, display, and edit the file.

To edit a sorted file, follow the procedure "Modifying an Existing File" in the "EDITOR" section. In Step 2 (enter TEXT or T, followed by the file's name), use the output file name you assigned. For example, the output file name in Example 5-10 is SFFINAL.

Using the MERGE Program to Combine Sorted Files

You can merge two or more sorted files into a single file with the MERGE program. For example, you could merge a company's employee file for the San Francisco branch with that of the Los Angeles branch for a complete employee list.

NOTE

To successfully merge files, they must have the same record format. That means all fields must start at the same character position and be the same length.

To merge two sorted files:

1. Enter `RUN MERGE.PUB.SYS` to start the subsystem.

   The system displays the chevron (>) prompt.

2. Enter `INPUT`, followed by the names of the files to be merged (input files).

   ```
   >INPUT input file1, input file2
   ```

3. Enter `OUTPUT`, followed by the name of the file that will contain the merged material (output file).

   ```
   >OUTPUT output file
   ```

4. Enter `KEY` and specify the starting position of the field you want to merge, followed by the number of characters in the field to be merged.

   For example, to merge by the fourth field in Example 5-9, enter `KEY 35, 13`.

5. Enter `END` to start the MERGE operation.

   The system displays a number of statistics and the colon (:) prompt. This indicates that the system completed the MERGE operation.

Example 5-11 shows how merge, by last name, two sorted employee files. The employee file SFFINAL was created in Example 5-10; the file LAFINAL was created in the same record format for the company's Los Angeles employees. The file that contains the merged file is called COMFINAL. Example 5-11 also shows how to display the merged file with the EDITOR subsystem.

```
:RUN MERGE.PUB.SYS
>

>INPUT SFFINAL, LAFINAL

>OUPUT COMFINAL

>KEY 1, 10

>END
:

:EDITOR
/

/T COMFINAL

/LIST ALL

Alter     Denise    62 Ocean Way         Los Angeles     CA   394-38-4291   01
Chu       Barbara   144 Hollywood Dr.    Burbank         CA   593-11-2837   04
Harvey    Bristol   1255 Page St.        San Francisco   CA   134-65-6894   02
Ing       Bobby     24-16th Ave.         Daly City       CA   655-43-7845   01
Jones     Tom       222 Cole St.         San Francisco   CA   263-15-0208   02
Kerney    Pete      77 Sunset Strip      Los Angeles     CA   655-23-8563   04
Martinez  Jennifer  1680 Boyd Rd.        San Francisco   CA   225-26-4467   04
Stuart    Annie     165 Fairview Dr.     Oakland         CA   225-48-5834   02
Turner    Rachel    64 Brook Blvd.       Los Angeles     CA   834-43-2531   02


/EXIT
:
```

Example 5-11. Merging and Displaying Two Sorted Files

STORE/RESTORE

The STORE/RESTORE subsystem is described in detail in *System Operation and Resource Management Reference Manual* (32033-90005), and *Storing and Restoring Files* (32033-90133). If you need to create a tape for a system other than the HP 3000, consult with your System Manager.

This subsystem has two basic commands-- :STORE and :RESTORE. The :STORE command is used to copy files from disc to magnetic tape. This is done primarily for two purposes: to "back up" files in case of a system crash or to send tapes containing data to other locations. Files copied to tape with the :STORE command can be copied back to disc only with the :RESTORE command.

Although there are other ways to copy files from disc to tape, always use the :STORE command for backups and for tapes to be transported. The :STORE command copies files in a format that is optimal for these uses. One of the most useful features of the :STORE command is that it creates a directory of files on that tape.

The :RESTORE command copies back to disc any files that were stored on tape with the :STORE command. However, since the original disc files are not affected by the process, you need to do this only under special circumstances.

The most obvious reason to use the :RESTORE command is to recover files after a system crash. Since the original disc files may be damaged or lost, the operations staff can recover the files from the STORE tape. Another reason to restore tapes is to incorporate changes made to a file. For example, if you send out a document on tape for review, the person reviewing it may make certain changes. By using the :RESTORE command, you can copy the edited file to your own system.

NOTE

If you copied files from disc to tape with FCOPY or another program, you need to use FCOPY to copy the file back to disc. The :RESTORE command works only on files copied with the :STORE command.

There are two types of magnetic tape devices: a cartridge tape drive, which uses cartridge tapes, and a reel-to-reel tape drive, which uses reel tapes. A few systems use removable discs in addition to magnetic tape. These discs are called "private volumes." In these systems, the STORE/RESTORE subsystem copies to and from these media.

About the Tape

The System Operator will mount the tape and perform the other tasks necessary to use the STORE subsystem. You should find out, however, whether or not you have to give a tape to the Operator. In either case, the Operator assigns a name to your tape. Make note of this name; you will need to use it when you copy your files with the :STORE command. To find out more about the Operator's procedures, consult the *Guide for the New System Operator* (32033-90021).

Using the STORE Subsystem

Copying a file to tape is essentially a two-step process, using the :FILE and :STORE commands.

To "store" a file to tape:

1. Enter :FILE, followed by the tape name assigned by the operator. Then specify the backup device, as shown below.

   ```
   :FILE tapename;DEV=TAPE
   ```

2. Enter :STORE, followed by the file name of the file to be copied, and a "backreference" to the tape name defined in Step 1, as shown below:

   ```
   :STORE filename ;*tapename
   ```

NOTE

Step 1 in the STORE procedure is a "file equation." File equations, as well as the concept of a backreference, are explained in more detail in Chapter 7, "Working with Files".

Example 5-12 shows how to copy a file named MYFILE (which happens to be in your group and account) to a tape named MYTAPE.

```
:FILE MYTAPE ;DEV=TAPE

:STORE MYFILE ;*MYTAPE

STORE/RESTORE; VERSION 2 (C) 1981   HEWLETT-PACKARD CO.
TUE, JULY 7, 1987, 2:57 PM

FILES STORED:     1
```

Example 5-12. Copying a Single File to Tape with the :STORE Command

Example 5-13 shows how to copy all files in your group (which is called FINANCE) to the tape MYTAPE. Note the use of the wild card character @ (see "Finding Files" in Chapter 7).

```
:FILE MYTAPE;DEV=TAPE

:STORE @.FINANCE;*MYTAPE

FILES STORED:     8
```

Example 5-13. Copying Several Files to Tape with the :STORE Command

The Execution Options

In addition to the basic procedure, you can also specify several optional parameters that define the execution options of the :STORE command. For a complete list of options, consult the *System Operation and Resource Management Reference Manual* (32033-90005). This section explores three execution option parameters: SHOW, PROGRESS, and PURGE.

The SHOW Parameter

If you enter the basic :STORE command as shown in Examples 5-12 and 5-13, the system executes the command and displays the total number of files stored and not stored.

The SHOW parameter displays additional information. It prints a message on the terminal as each file is stored, including the file name, group name, account name, logical device number, file size (in sectors), and the file code. A typical message is shown in Example 5-14.

If problems occur, the SHOW parameter also displays messages, such as DEVICE UNAVAILABLE (FSERR55), FAILED TO OPEN FILE MYFILE (FSERR52), and others. In most cases, the operations staff will need to resolve these problem. If there is anything you need to do, they will let you know.

Example 5-14 shows how to use the SHOW parameter to display file names as they are stored.

```
:FILE MYTAPE;DEV=TAPE

:STORE @.FINANCE;*MYTAPE; SHOW

file name  GROUP    ACCOUNT  LDN    ADDRESS    REEL   SECTORS  CODE

SAMPLE     .FINANCE.ACCT     3%0114354    1          257      FIG
MYFILE     .FINANCE.ACCT     1%00120467   1          50       RASTR
RED        .TAPE.GVT         2%001253521  1          544      RASTR

FILES STORED:       3
```

Example 5-14. Using the SHOW Parameter in the :STORE Command

The PURGE Parameter

Normally, when you use the :STORE command to copy a file from disc to tape, the original disc file is not affected. However, there may be circumstances when you want to erase the original copy on the disc. For example, if you know that file will be used infrequently, you can conserve disc space by erasing the file as you store it on tape for future reference.

WARNING

Use the PURGE parameter with care. If the tape is defective or if it is destroyed, the data is lost.

Example 5-15 shows how to erase the original disc file by adding the PURGE parameter.

```
:FILE MYTAPE;DEV=TAPE

:STORE @.FINANCE;*MYTAPE;PURGE

STORE/RESTORE; VERSION 2 (C) 1981   HEWLETT-PACKARD CO.
TUE, JULY 7, 1987, 2:57 PM
```

Example 5-15. Using the PURGE Parameter in the :STORE Command

The PROGRESS Parameter

When you copy large numbers of files you may want periodic progress messages. Unless defined, the PROGRESS parameter displays a progress message every minute. You can also specify different time intervals.

Example 5-16 shows how to display a progress message every five minutes.

```
:FILE  MYTAPE;DEV=TAPE

:STORE @.FINANCE;*MYTAPE; PROGRESS=5

STORE/RESTORE; VERSION 2 (C) 1981   HEWLETT-PACKARD CO.
TUE, JULY 7, 1987, 2:57 PM
```

Example 5-16. Using the PROGRESS Parameter in the :STORE Command

Using the RESTORE Subsystem

To recover a file from tape, you essentially reverse the STORE procedure. First, give the tape that contains your files to the Operator to get ready. Then enter the :RESTORE command, as shown below. Note that the order in step 2 is reversed from that in the :STORE command.

If you use the SHOW parameter with the :RESTORE command, the system displays a message each time it restores a file, as well as messages that indicate problems. You can also use the PROGRESS execution options to keep track of your progress.

To restore a file from tape:

1. Enter :FILE, followed by the tape name assigned by the Operator. Then specify the backup device (that is, the tape or private volume), as shown below.

   ```
   :FILE tapename;DEV=TAPE
   ```

2. Enter :RESTORE, followed by the "backreference" to the tape name, followed by the file name.

   ```
   :RESTORE *tapename; filename(s)
   ```

NOTE

Step 1 in the RESTORE procedure is a file equation. File equations, as well as the concept of a "backreference", are explained in more detail in Chapter 7, "Working with Files".

Example 5-17 shows how to restore all files belonging to the group FINANCE, which were stored to the tape MYTAPE in Example 5-13.

```
:FILE MYTAPE;DEV=TAPE

:RESTORE *MYTAPE; @.FINANCE

FILES RESTORED:   3

STORE/RESTORE; VERSION 2 (C) 1981   HEWLETT-PACKARD CO.
TUE, JULY 7, 1987, 2:57 PM
```

Example 5-17. Using the :RESTORE Command

FCOPY

The FCOPY subsystem can copy files from any input device to any output device, as shown in Figure 5-6. It also offers a convenient way to make multiple copies of files, to transfer files between groups or accounts on the same system, or to transfer data from one medium to another. For a complete list of capabilities, see the *FCOPY Reference Manual* (03000-90064).

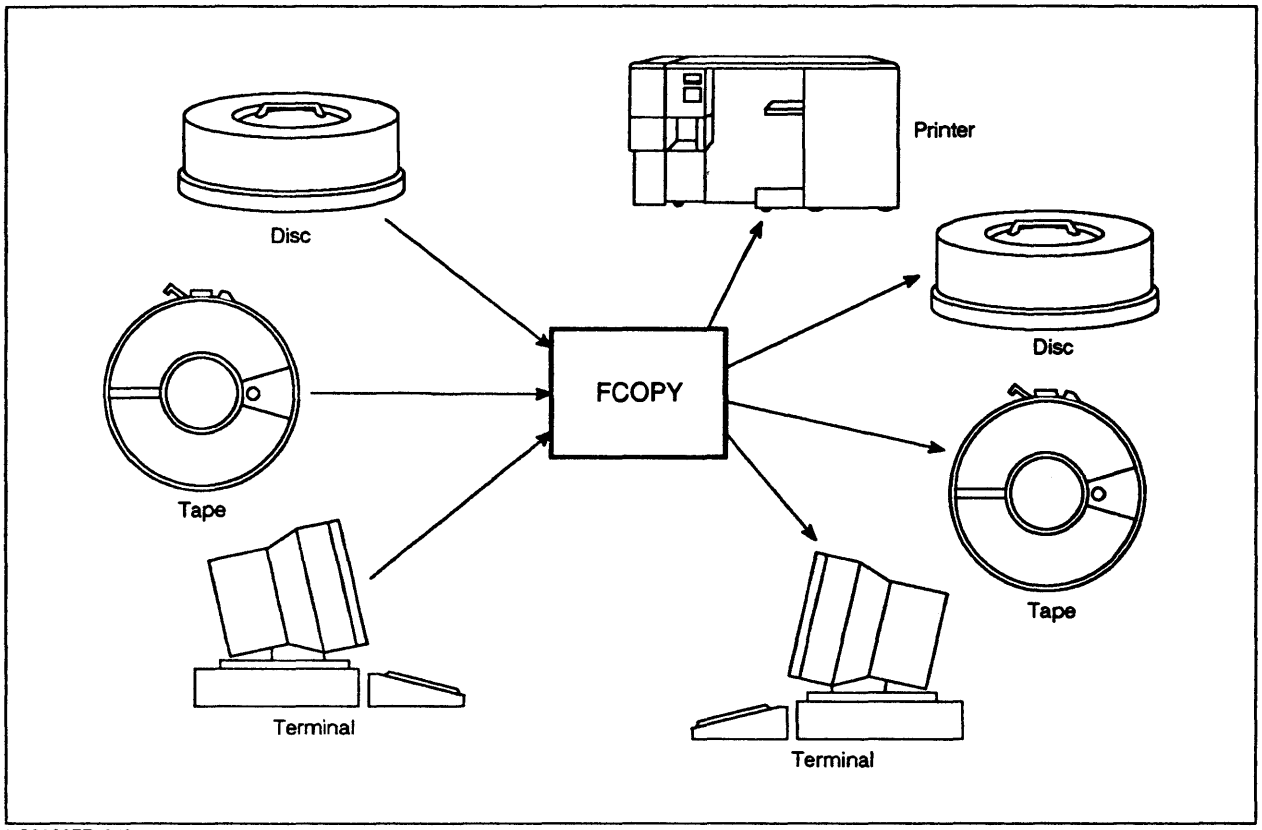


Figure 5-6. The FCOPY Subsystem

To copy files from disc to tape for backup or transport, use the STORE/RESTORE subsystems. Although you could use FCOPY to do this, the STORE/RESTORE subsystems have several advantages, including a feature that provides a directory of files on the resulting tape.

FCOPY, on the other hand, is extremely flexible. The list below just shows some of the things it can do. For a complete list and description of functions, refer to the *FCOPY Reference Manual* (03000-90064).

- Print a file.
- Show the contents of any disc file, in a variety of formats
- Copy a disc file from one group or account to another.
- Compare the contents of two files.
- Translate files from one computer code to another.

NOTE

The FCOPY subsystem displays three types of messages: status, error, and warning. The messages shown in this section are examples only. Appendix A in the *FCOPY Reference Manual* (03000-90064) lists all FCOPY messages.

This section explains only how to duplicate an existing file, how to copy files between groups or accounts on the same system, how to compare two files, and how to print with the FCOPY subsystem.

Using the FCOPY Subsystem

In general, to perform any FCOPY function, specify a "from" file that contains the data you want to copy and a "to" file to which the data is copied. You can then specify additional FCOPY functions. The basic FCOPY syntax looks like this:

```
:FCOPY FROM=source filename.groupname.accountname;TO=target filename;
function
```

NOTE

If the contents of the source file are in a different group or account, the file's creator may need to use the :RELEASE command to temporarily void the security provisions that apply to that file. Procedures are listed in Chapter 6, "Basic Account Structure."

Example 5-18 shows how to perform one of the simplest FCOPY functions; creating a new file by copying an existing file. Since the new file is in the same account as the file to be copied, the group and account names are not necessary.

```
:FCOPY FROM=MYFILE; TO=MYFILE2;NEW

 HP 32212A.03.23  FILE COPIER (C)  HEWLETT PACKARD CO.  1984

EOF FOUND AFTER RECORD 2664

2663 RECORDS PROCESSSED  ***  0 ERRORS

END OF SUBSYSTEM
:
```

Example 5-18. Creating a New File with the :FCOPY Command

Copying Files between Accounts

If the contents of the source file are in a different group or account, the file's creator may need to use the :RELEASE command to temporarily void the security provisions that apply to that file before it can be copied with the FCOPY subsystem. Security provisions are described in more detail in Chapter 6, "Basic Account Structure".

To copy files between groups or accounts with the FCOPY subsystem:

1. Ask the creator of the file to temporarily remove the security provisions that apply to the file you want to transfer.

   ```
   :RELEASE filename
   ```

2. Log on to account you want to transfer to.

   ```
   :HELLO username.accountname (passwords, if any)
   ```

3. Copy the file

   ```
   :FCOPY FROM=filename.groupname.accountname;TO=filename; NEW
   ```

   or

   ```
   :FCOPY
   >FROM=filename.groupname.accountname;TO=filename; NEW
   >EXIT
   ```

4. (Optional) enter LISTF to check to make sure the file was copied.

   The system displays all file names in your account. It should include the file name of the file you copied.

5. Tell the creator of the file to replace the security lock on the file you transferred.

```
:SECURE filename
```

Example 5-19 shows how to use the FCOPY subsystem to copy the file STATS from the FINANCE account (group AUDIT, user WATSON) to the MARKET account, (group RESEARCH, user SHERLOCK).

```
:HELLO WATSON.FINANCE                                    (user 1)

        *****          WELCOME TO SAMURAI          ******

:RELEASE STATS
:
```

```
:HELLO SHERLOCK.MARKET                                   (user 2)

        *****          WELCOME TO SAMURAI          ******

:FCOPY
>
>FROM=STATS.AUDIT.FINANCE;TO=STATS2;NEW

New file created - STATS2.SHERLOCK.MARKET
Succeeded

>EXIT
```

```
:HELLO WATSON.FINANCE                                    (user 1)

        *****          WELCOME TO SAMURAI          ******

:SECURE STATS
```

Example 5-19. Using FCOPY to Copy between Accounts on the Same System

Comparing File Contents with FCOPY

The COMPARE function lets you compare the contents of two files without changing either file, as shown below.

To compare the contents of two files:

Copy the file with the :FCOPY command and add the function COMPARE.

```
:FCOPY FROM=sourcefile; TO=targetfile;COMPARE
```

Example 5-20 shows how to compare the contents of two files in the same account.

```
:FCOPY  FROM=STATS.AUDIT.FINANCE;TO=STATS2;COMPARE

HP 32212A.03.23  FILE COPIER (C)  HEWLETT PACKARD CO.  1984

EOF FOUND IN FROMFILE AFTER RECORD 52

53 RECORDS PROCESSED  *** 0 ERRORS

END OF SUBSYSTEM
:
```

Example 5-20. Using the :FCOPY Command to Compare File Contents

Printing Files with the FCOPY Subsystem

Instead of going into the subsystem that created a file to print it, you can use the FCOPY subsystem. This is especially useful if you want additional control over the printing process, such as specifying a printer, the number of copies, and so on.

To use the FCOPY subsystem, you would first enter a file equation that defines the device to which you want to copy the file. (For further information about file equations, see Chapter 7, "Working with Files".)

To print a file with FCOPY:

1. Enter FILE, followed by a file equation that defines the printer and its specifications.

   ```
   :FILE filename1;DEV=device specifications
   ```

2. Enter FCOPY, followed by the file name of the file to be copied and a "backreference" to the file equation.

   ```
   :FCOPY FROM=filename2;TO=*filename1
   ```

Example 5-21 shows how to use a file equation and the FCOPY subsystem to print a file on the laser printer.

```
:FILE PRINTER; DEV=PP
:FCOPY  FROM=MYFILE;TO=*PRINTER
```

Example 5-21. Using the FCOPY Command to Print a File

SPOOK

The SPOOK subsystem is described in detail in the *MPE V Utilities Reference Manual* (32033-90008).

The SPOOK subsystem works in sessions only. It makes it possible to look at, manipulate, or delete "spoolfiles," which are files waiting to be printed. As a general user, you probably have access only to those spoolfiles you created.

About Spoolfiles

As you know, only one user at a time can use printers and tape drives. For that reason, those devices are called "nonsharable devices." Trying to catch a printer when it is free could be very frustrating, similar to trying to call a number that is always busy.

The spooler program makes the user's life easier. It creates spoolfiles of all files destined for printers, tapes, or other nonsharable devices. These spoolfiles are exact copies of the disc files. However, they exist only for the time they wait their turn to be processed.

About SPOOK Tapes

In addition to manipulating spoolfiles, the SPOOK subsystem can create "spook tapes". These tapes can save spooling time, making it easy to transfer files already spooled between printers on different systems or to transport them to other sites. For further information, check the *MPE V Utilities Reference Manual* (32033-90008).

The SPOOK Commands

Like EDITOR, the SPOOK subsystem has its own set of commands, as well as its own HELP facility. For a complete list of commands, as well as all the options available by using parameters, check the *MPE V Utilities Reference Manual* (32033-90008). This section covers only the basic commands (with default parameters) summarized in Table 5-3:

Table 5-3. Summary of Selected SPOOK Commands

Command	Shortcut	Function
SPOOK		Starts the SPOOK Subsystem.
TEXT	T	Opens the spoolfile and prepares it for the >LIST or >FIND commands.
LIST	L	Displays all or selected lines in file (used after TEXT command).
FIND	F	Locates specific character strings.
APPEND	AP	Adds all or part of a spoolfile to another spoolfile.
ALTER	A	Changes output priority, number of copies requested, or the device that will print the file.
PURGE	P	Deletes the output spoolfile from the system.
SHOW	S	Lists the characteristics of input and output spoolfiles.
XPLAIN	X	Shows syntax and usage for SPOOK commands. Can also reach MPE's HELP facility.
EXIT	E	Exits the SPOOK subsystem.

Using the SPOOK Subsystem

As you can see from Table 5-3, some SPOOK commands are similar to EDITOR's subcommands. Other SPOOK commands let you change your mind about which printer to print on, how many copies to print, and so on.

Getting Started

SPOOK can access any files in the READY state. Before you can use the SPOOK commands, you need to know the file's state, as well as the output file number assigned by the spooler. The >SHOW command displays this information.

1. Start the SPOOK subsystem.

   ```
   :RUN SPOOK5.PUB.SYS
   ```

   In response, the system displays the chevron (>) prompt.

2. Enter SHOW to display information about the file.

   ```
   >SHOW
   ```

Example 5-22 shows how the user LEE.ART logged on, started the SPOOK subsystem, and displayed the information necessary to use it.

The screen shown in Example 5-22 is the "short" format. As you can see, the system has assigned each file an output number, as indicated by the O that precedes the number. It also shows the session or job number, the name of the spoolfile, its state, and its creator (owner).

```
:HELLO LEE.ART
:RUN SPOOK.PUB.SYS
>
>SHOW

     #File      #JOB              FNAME      STATE      OWNER
     #O263      #S111             EDTLIST    READY      LEE.ART
     #O264      #S111             EDTLIST    ACTIVE     LEE.ART
     #O265      #S111             EDTLIST    READY      LEE.ART
     #O266      #S111             EDTLIST    READY      LEE.ART
```

Example 5-22. Sample SHOW Screen

NOTE

If the system displays a chevron (>) prompt instead of a display similar to Example 5-22 in response to the >SHOW command, it means that there are no spoolfiles to be accessed. Any files you may have sent to the printer have already been processed.

Displaying Spoolfiles

You can look at any spoolfiles in the READY state. Once a file is displayed, you can use the SPOOK commands to manipulate the file.

To display spoolfiles:

1. Enter TEXT and the file's output number (shown in the >SHOW display) to open the file.

   ```
   >TEXT filenumber
   ```

2. Enter LIST, followed by the number of lines you want to display.

   ```
   >LIST linenumber(s)
   ```

For example:

- L ALL displays all lines in the file.
- L 3 displays line 3.
- L 3/9 displays lines 3 through 9.

Example 5-23 shows how to display lines 25 through 31 in one of the spoolfiles.

```
>TEXT 263
>

>LIST 25/31

   25     Whoever said Hollywood movies were the
   26     opiates of the masses had it wrong.
   27     Anyone who goes to summer movies can tell
   28     you that Hollywood's pushing stimulants.
   29     Every plot must have 18 cliffhangers,
   30     myriad car chases and lots of fire-
   31     power--and these are comedies we're
```

Example 5-23. Displaying a Spoolfile

NOTE

You do not need to specify the "O" (for OUTPUT) before the file number assigned by the spooler. However, if you do use it, be sure it is an "O" and not a 0 (zero).

Locating Data in Spoolfiles

Once a spoolfile is displayed, you can find particular text by defining a "character string." A character string may consist of any number of characters enclosed in quotation marks. For each character string found, the system displays the line number and the text in that line.

To find a character string:

1. Display the spoolfile

   ```
   >TEXT filenumber

   >LIST line number(s)
   ```

2. Enter FIND, followed by the text you want to locate, in quotation marks, and the section of the file you want to search.

   ```
   >FIND "character string"; line number(s)
   ```

Example 5-24 shows how to search lines 1 through 50 in spoolfile #O266 to locate the first instance of the word "default."

```
>TEXT 266
>

>FIND "default"; 1/50

   29     computer's decision is called "default."
```

Example 5-24. Locating Data in Spoolfiles

Adding to (Appending) Spoolfiles

You can also combine two spoolfiles or add parts of one spoolfile to another.

To append two files:

1. Enter APPEND, followed by the two file numbers and the range of text to be added.

   ```
   >APPEND filenumber, filenumber, ALL or desired line numbers (range)
   ```

2. Enter APPEND END to exit from Append mode.

   ```
   >APPEND END
   ```

Example 5-25 shows how to add lines 230 through 248 in file #263 to file #266, using the abbreviated command.

```
>AP 263, 266; 230/248

>AP END
>
```

Example 5-25. Adding Data to Spoolfiles

Deleting Spoolfiles

You can stop a file from printing by deleting its spoolfile.

To delete a spoolfile:

Enter PURGE, followed by the file number you want to delete.

```
>PURGE filenumber
```

Example 5-26 shows how to delete spoolfile #O264, using the abbreviated command.

```
>P 264
>
```

Example 5-26. Deleting Spoolfiles

Changing the Printer, Number of Copies

The >ALTER command lets you change your mind about which printer to use or how many copies to print, as well as the file's output priority.

To change the printer:

Enter ALTER or A, followed by the spoolfile's number and the new device designation.

```
>ALTER filenumber; DEV=device designation
```

To change the number of copies printed:

Enter ALTER or A, followed by the spoolfile's number and the number of copies.

```
>A filenumber; COPIES=number of copies
```

Example 5-27 shows how to redirect file #O266 to a laser printer instead of the line printer, and to print three copies.

```
>ALTER  O266;DEV=PP;COPIES=3
>
```

Example 5-27. Changing a Spoolfile's Output and Number of Copies Printed

Using SPOOK's HELP Facility

Unlike MPE's HELP facility, the SPOOK HELP facility does not display definitions of the commands or examples of their use. Instead, it simply shows a list of SPOOK commands and the parameters that apply to those commands, as shown in Example 5-28.

To use the SPOOK HELP Facility:

Enter XPLAIN.

```
>XPLAIN
>

SHOW            [  USER  [  .ACCOUNT  ] ] [ ; [O] [I] [O]  ]
SHOW            DEVICEFILEID  [  ,  DEVICEFILEID  ] . . . .
TEXT            DEVICEFILEID
LIST            [  RANGE  ]
FIND            [ @ ]  [ "STRING" ]  [ , FRANGE ]
MODE            [ OPTION [ , OPTION ] . . . ]
                OPTION = WIDTH / CONTROLS
ALTER <DFID [ ,DFID, . . . ]]. [ , OPTION [ , OPTION ] . . . . ]
ALTER <USER [.ACCOUNT]  > [ ; OPTION [ , OPTION ] . . . . ]
                OPTION = PRI / COPIES / DEV
PURGE           DEVICEFILEID [ , DEVICEFILEID ] . . .
HELP
RUN             PROGRAMfile name [ .GROUP [ .ACCOUNT]  ]
KILL            << SON PROCESS >>
QUIT            <<TERMINATE>>
COPY            [RANGE [ ,file name]]
COPY            [DFID [,DFID [,...]] ;] [RANGE [ ,file name]]
COPY            [USER [.ACCOUNT] ;]   [RANGE [ ,file name]]
APPEND          [DFID [,DFID [,...]] ;] [RANGE [,file name]]
APPEND          [USER [ACCOUNT] ;]  [RANGE [,file name]]
                [END  ]
```

Example 5-28. Using the SPOOK HELP Facility

6

Basic Account Structure

Chapter Overview

This chapter describes MPE's account structure. It explores its primary components and functions, including:

- The account structure's components
- Rules for naming account components and files
- A review of the directory and its function
- The account structure's three main functions, including billing, organization, and security
- The use of security passwords for accounts, groups, and users
- The use of security lockwords for files
- An introduction to file access permissions
- Checking file access permissions
- An introduction to capabilities
- Checking capabilities

Account Structure Components

Just as filing cabinets serve to organize files and their owners, some sort of structure is necessary to organize files and users on the HP 3000. This structure is MPE's account structure. It includes four basic components: accounts, groups, users, and files. Each of these components can be assigned a password that keeps unauthorized users from accessing it. For example, accounts can have account passwords, groups can have group passwords, and so on. These security measures are described in detail later in this chapter.

For an overview of how the four basic components interrelate, see Figure 6-1. As you can see, users are organized into groups, which in turn make up accounts. Information is organized into files, which are created by individual users.

Accounts (TECHNLGY, MARKTING, SYS, and so on) are listed horizontally. Groups (RESEARCH, SALES, RECORDS, and so on) are stacked vertically under their accounts. Users (KEVIN, BRENDA, BARNEY, etc.) are stacked vertically under their groups, which is known as the "home group."

Accounts

Accounts are the basic structure for organizing users and information in your system. Groups, users, and information (files) "belong" to accounts. When you log on, you must specify your account.

Each account has an Account Manager responsible for establishing and maintaining the groups and the users within the account. The duties of the Account Manager are detailed in *MPE V Accounting, Billing and Security* (32033-90136).

Groups

Groups further organize users and files within accounts. Unless you specify another group when you log on, the system automatically logs you into your "home" group (assigned by the Account Manager). Although users can log on to any group in their account, they usually work in and store files in their home group.

Note that each account also includes a group named PUB (for public). This group contains information useful to everyone in that account, such as shared files, programs, and so on.

Users

Users are the people who do their work on the HP 3000. However, a user isn't necessarily a single person. An entire department might be considered a user, for example.

Files

Files belong to groups. Files store the information created and maintained by users, including programs, spreadsheets, memos, manuals, and so on.

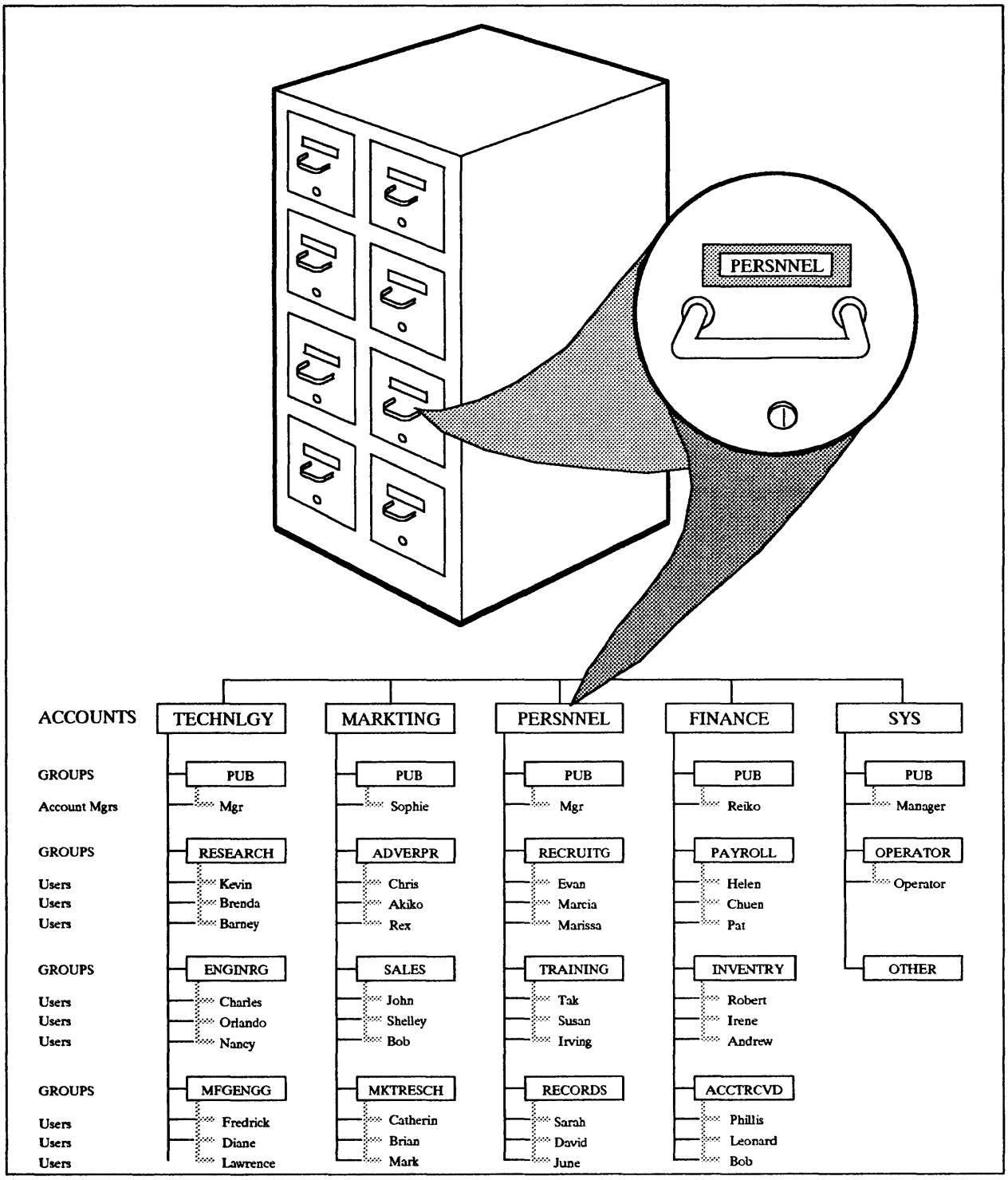


Figure 6-1. Account Component Relationships

Naming Conventions

Except for the system-assigned account SYS, all names related to the account structure are assigned by someone. For example, the Account Manager assigns names to accounts, groups, and users. These names are usually related to the account's or group's function within the organization, such as Finance, Marketing, or Research.

Users assign names to the files they create. Like account or group names, you may also want to choose file names that identify the files contents. For example, you could name a file of customer orders "falcon," or anything else that follows the rules listed below. However, the file name "orders" lets you (and anyone else who needs access to your files) find it quickly among other files.

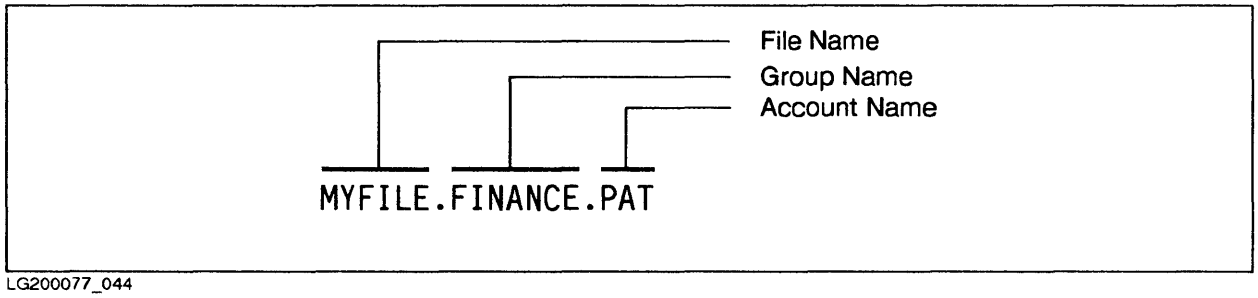
Naming Rules

Account components include accounts, groups, users, and files, as well as any passwords or lockwords associated with these components. Component names must follow these rules:

- They must be eight characters or less.
- They must begin with a letter.
- They may not contain special characters or spaces.

Fully Qualified File Names

The HP 3000 stores permanent files in a special format, called the "file address" or "fully qualified file name." This format consists of the file name, the group name, and the account name, as shown in Figure 6-2.



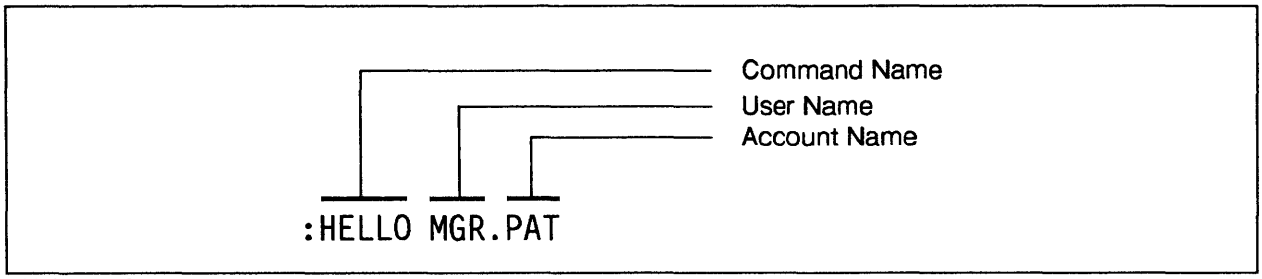
LG200077_044

Figure 6-2. The Fully Qualified File Name

Unless security provisions prevent it, you must use the fully qualified file names to list, run, or modify files not in your group or account. To copy files between accounts and between systems, you also need to use the fully qualified file name.

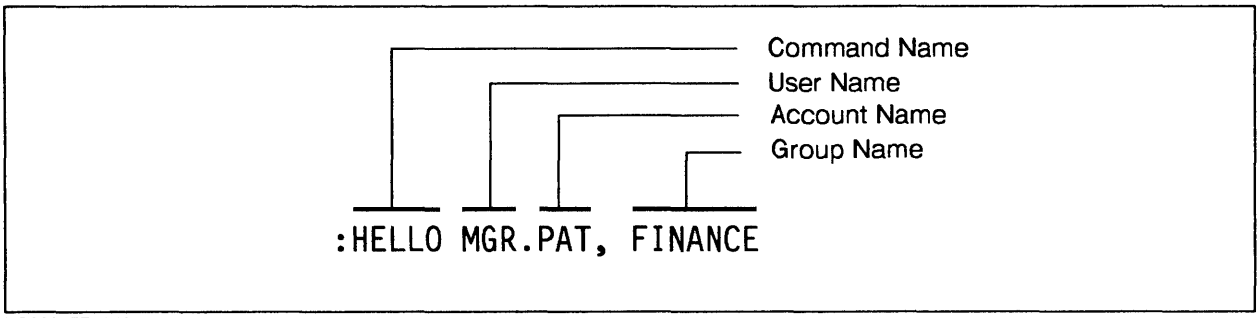
The Group Name

As you know, the :HELLO command, followed by the user name and the account name, logs you on to the HP 3000. Specifying a group is optional. Figure 6-3 shows the basic command; Figure 6-4 shows the command that specifies a group other than the home group.



LG200077_045

Figure 6-3. Basic Logon Sequence



LG200077_046

Figure 6-4. Logon Sequence with Optional Group

The Directory

MPE's directory, which is located on the system disc, stores all the information pertaining to the account structure. It knows which users belong to which group, which groups are owned by which accounts, as well as any passwords associated with users, groups, and accounts.

When you log on with the :HELLO command, MPE checks to make sure the user name and the account name you specified exist. It also determines what home group is associated with that user name. In addition, MPE checks for passwords.

If the information you entered with the :HELLO command matches that in the directory, you are logged on. The system displays a : (colon) prompt and you can use the computer, discs, printers, and other system resources.

If the information does not match, an error message appears on the screen. If that happens, check punctuation, spelling, and whether any passwords are assigned to that account, group, or user.

Functions of the Account Structure

MPE's account structure performs several basic, necessary functions. Like thousands of Hewlett-Packard users, you can log on and successfully use the HP 3000 without knowing about

your logon user, group, or account capabilities. However, understanding and being able to effectively use the account structure makes you a more flexible and effective user.

The account structure's three main functions include:

- Organization
- Billing
- Security

Organization

The HP 3000's vast resources can process huge quantities of information by hundreds of users. Obviously, some structure is necessary to organize it all. The account structure provides this function by organizing information into files, and users into groups and accounts, as illustrated in Figure 6-1.

Billing

This account structure also allows MPE to track the usage of system resources by users, groups, and accounts. In this way, management can monitor system usage and charge the appropriate accounts, if desired. The account structure can keep track of the following resources:

- CPU time, which is the time the computer's CPU is used, measured in seconds.
- Connect time, which is regular clock time, measured in minutes. It includes the time from when a user logs on to when a user logs off.
- Disc storage space used.

NOTE

In addition to these tracking resources, MPE also provides a system logging facility.

Security

The account structure's most important function is security. In addition to protecting computer equipment from theft and damage, it is essential to protect the information on the system. Much of the information stored on computers is highly confidential; only authorized people should have access to it.

In addition, it is important to protect data from being tampered with or destroyed. This means protecting the system from the potential thief, as well as the person who simply makes a mistake.

MPE automatically assigns security provisions (defaults) at all levels. In addition, management can impose stricter security provisions at the account and group levels to which a file belongs. Users can also protect files they create by adding lockwords and changing the default file security provisions assigned by MPE.

In summary, MPE's account structure contains the following built-in security features:

- Passwords and lockwords
- File access permissions
- Capabilities

NOTE

This section includes step-by-step instructions for several tasks. To avoid repetition, assume the following:

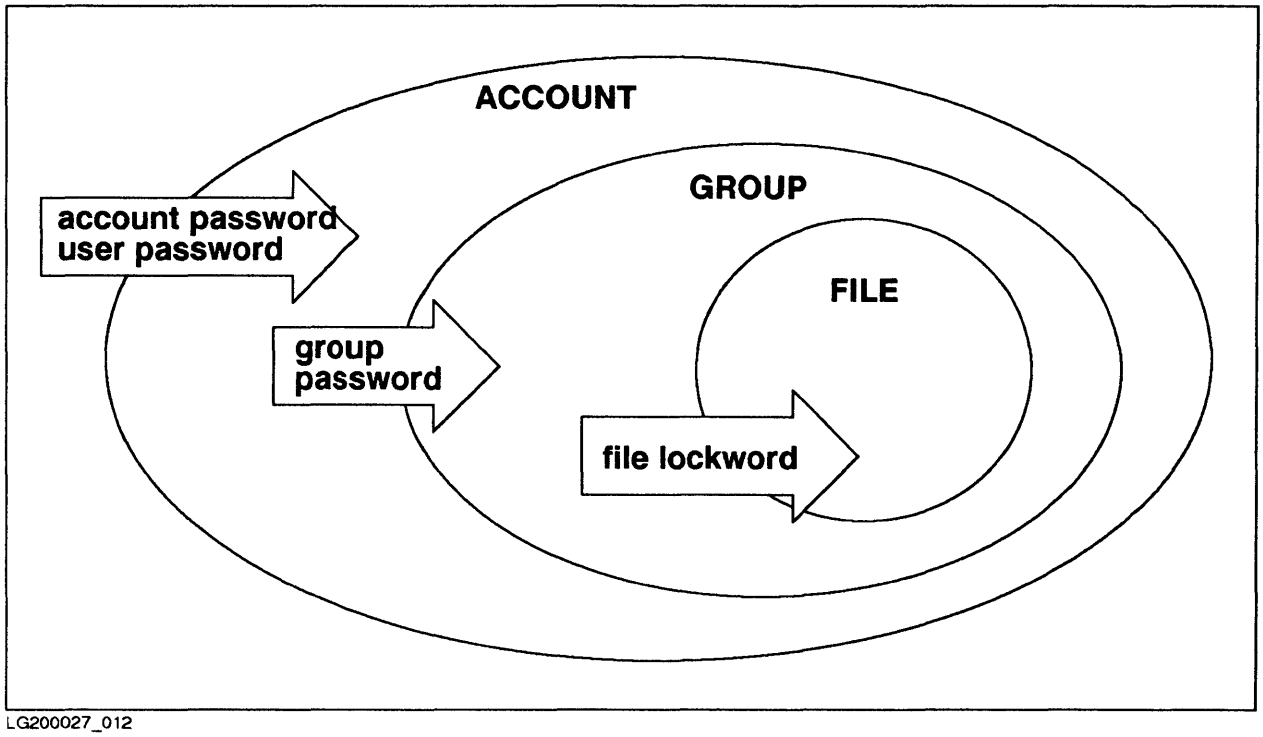
1. At each step, you enter information at the colon (:) prompt, unless specified otherwise.
2. After entering information, you press Return.

Passwords and Lockwords

Passwords limit access to accounts, groups, and users. The System Manager assigns the account password. The Account Manager assigns the group and user passwords.

Lockwords limit access to files. The user who creates the file assigns the lockword, if desired.

Figure 6-5 shows how passwords and lockwords relate to the account components.



LG200027_012

Figure 6-5. Passwords and Lockwords

If passwords were assigned for the account, group, or user, only users who know these passwords can access the files associated with that account, group, or user.

Although passwords could be included in the logon sequence, this is discouraged because the password would be displayed on the screen as it is entered. In fact, MPE may be programmed not to accept passwords embedded in the logon sequence.

When a password is required, the system prompts you for the password by displaying

```
ENTER ACCOUNT PASSWORD:
```

When you enter the password in response to this prompt, the letters are *not* displayed on the screen.

Your manager should tell you the necessary user and account passwords. Group passwords are necessary only when logging on to groups other than your home group.

Figure 6-6 shows the :HELLO command with every possible password. Note that passwords for each parameter are preceded with a slash delimiter.

NOTE

This illustration is for demonstration only. For security reasons, do not include passwords with the logon command by using the slash (/) character.

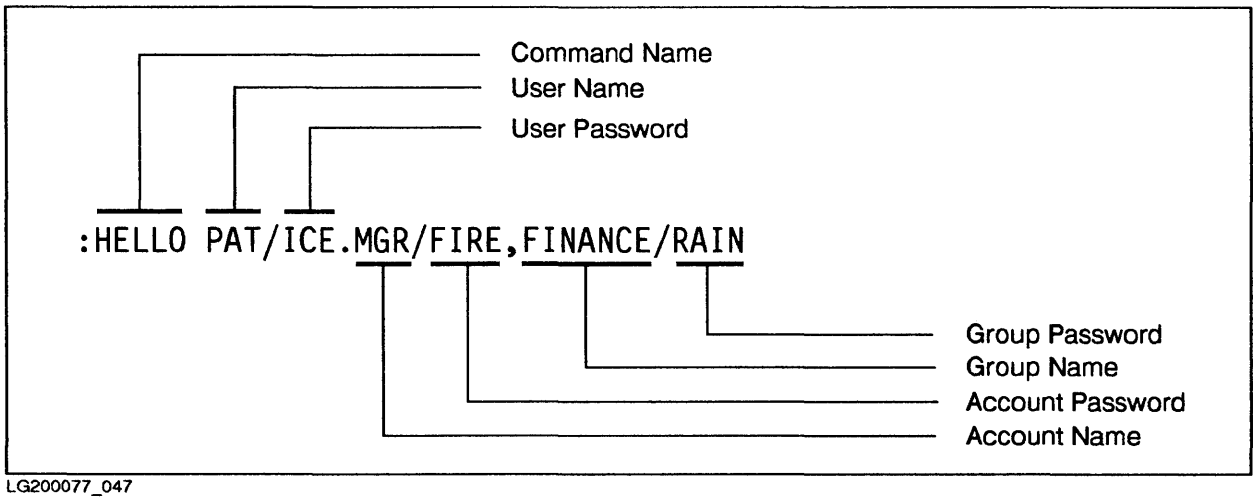


Figure 6-6. The :HELLO Command with Passwords

Changing Passwords

The System Manager has several additional options for increasing password security, including requiring user passwords, making user passwords expire at specified intervals, or requiring a minimum number of characters. These security precautions are described in detail in *MPE V Accounting, Billing, and Security* (32033-90136).

Your manager can tell you whether any of these options apply to your account. If you are expected to maintain your user password, follow the procedure below.

To create or change a password:

1. Enter `PASSWORD.`

   The terminal displays `ENTER OLD USER PASSWORD:`

2. If you have a password, enter that password. If you do not have a password, press Return.

   The terminal displays `ENTER NEW USER PASSWORD:`

3. Enter your new password.

The terminal displays ENTER NEW USER PASSWORD AGAIN:

4. Enter the new password once more.

   If you entered the password exactly as you did the first time, the terminal displays: PASSWORD WAS CHANGED SUCCESSFULLY.

   If the second new password did not match the first, the terminal displays: PASSWORD WAS NOT CHANGED. In this case, start again with Step 1.

Example 6-1 shows how to change the user password from BILBO to GANDALPH.

```
:PASSWORD

   ENTER OLD USER PASSWORD: BILBO

   ENTER NEW USER PASSWORD:  GANDALPH

   ENTER NEW USER PASSWORD AGAIN:  GANDALPH

   PASSWORD WAS CHANGED SUCCESSFULLY
```

Example 6-1. Changing User Passwords

Lockwords

Lockwords are assigned by the file's creator. They act as passwords for files, which must be supplied before the files can be accessed. Since nobody else knows your lockword and only the System Administrator can find out what it is, lockwords provide a very high level of security.

Creating Lockwords

When you choose a lockword for your file, keep in mind that they are easy to forget. One easy way to keep lockwords under control is to use a single lockword on all your files, but to change it periodically.

There are several ways to put lockwords on your files. In the EDITOR subsystem, you can assign lockwords when you create or modify files. Simply add a slash and the lockword to the file name when you use EDITOR's /KEEP command. This is the best way to add a lockword when you first create a file. (See the "EDITOR" section in Chapter 5 for detailed procedures on creating files.)

For existing files, the easiest way to add a lockword is to use the :RENAME command.

To add a lockword to an existing file:

Enter RENAME, followed by the file name and the lockword, separated by a slash.

```
:RENAME filename, filename/lockword
```

Example 6-2 shows how to add the lockword TGIF to the file MYFILE.

```
:RENAME MYFILE, MYFILE/TGIF
:
```

Example 6-2. Adding Lockwords to Existing Files

Changing Lockwords

You can also change lockwords with the :RENAME command.

To change a lockword:

Enter RENAME, followed by file name and existing lockword, and the file name with the new lockword.

```
:RENAME filename/lockword, filename/new lockword
```

Example 6-3 shows how to change the lockword TGIF on the file MYFILE to OGIM.

```
:RENAME MYFILE/TGIF, MYFILE/OGIM
:
```

Example 6-3. Changing Lockwords on Existing Files

Retrieving Lockwords

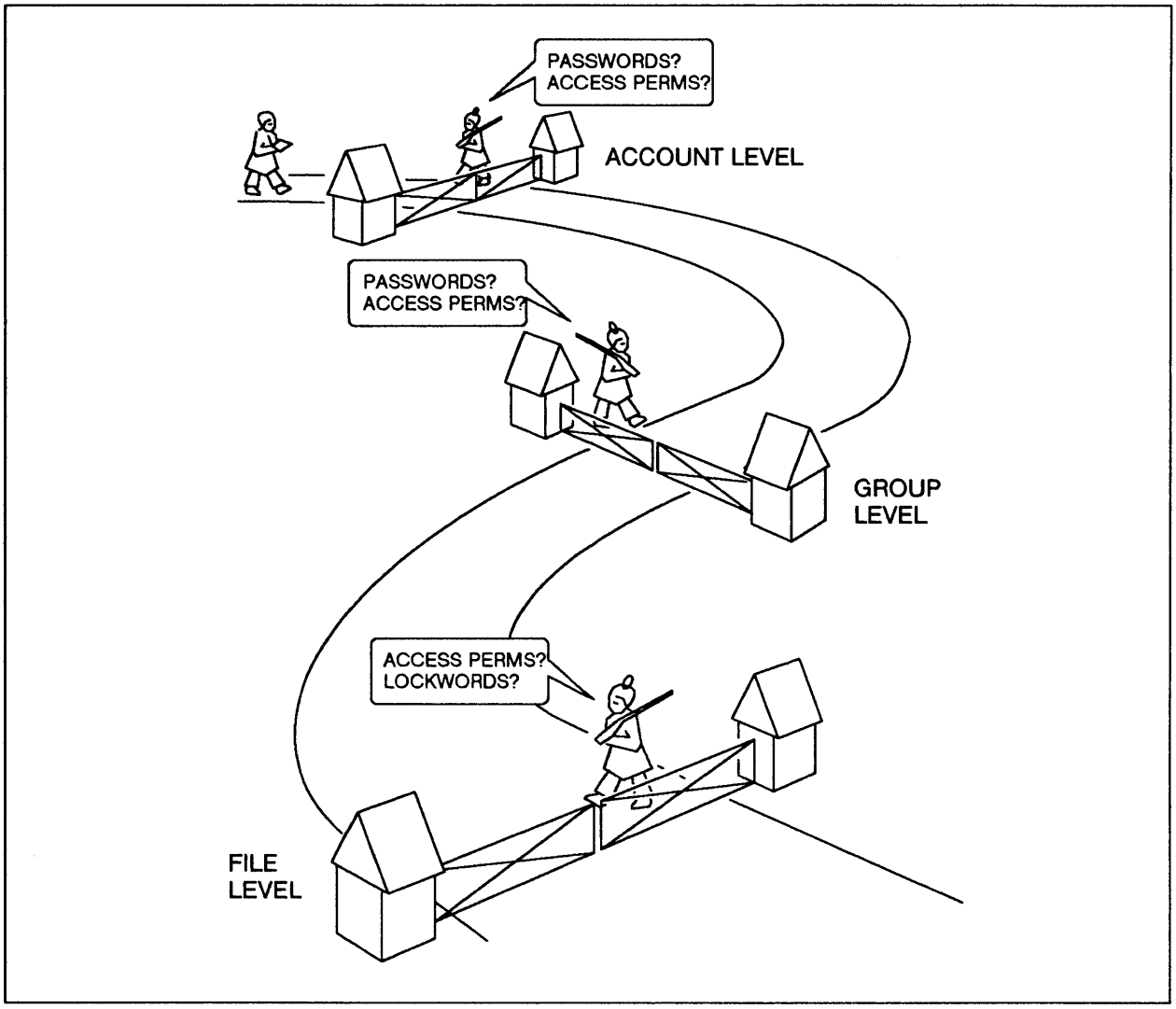
If you forget your lockword after a particularly good vacation, check with the System or Account Manager. They can use their special capabilities to find your lockword.

File Access Permissions

File access permissions determine who can do what to disc files; specifically, who can look at, change, and execute the contents of files. Only those permissions explicitly granted are in effect. File access permissions, which are summarized in Table 6-1, apply at the account, group, and file level.

File access permissions at the account and group level can be changed only by the Account Manager. Permissions at the file level can be changed by the user who created those files.

File access permissions have the following hierarchy: account-level permissions override group permissions, which in turn override file-level permissions. Therefore, a user who wants access to a file needs permission at all three levels. In a way this is similar to going through three access gates, as shown in Figure 6-7. For example, to look at a file in another account, you need any passwords that apply, as well as the appropriate access permission at the account, group, and file level.



LG200077_048

Figure 6-7. File Access Permissions at the System, Group, and File Levels

Unless otherwise specified, MPE automatically assigns all account and group users the file access permissions listed in Table 6-1.

Table 6-1. Default File Access Permissions

Access	Code	Allows User to:
READ	R	Read files.
APPEND	A	Add information to files. However, users can't change or delete information in the file.
WRITE	W	Add, delete, or change information. Users can also delete files with the :PURGE command.
LOCK	L	Prevent users from opening a file that is already open.
EXECUTE	X	Run programs stored in files with the :RUN command.
SAVE	S	Make files permanent, rename them, and create new files with the :BUILD command.

These permissions are assigned by equating each access permission with a user code. Table 6-2 shows a summary of the user codes.

Table 6-2. User Codes for File Access Permissions

User Type	Code	Description
Any user	ANY	Any user logged on to the account to which the file belongs.
Account Member	AC	Any user authorized to access the account to which the file belongs.
Group Librarian	GL	Users with the capability to manage files within the file's home group.
Account Librarian	AL	Users with the capability to manage files within the account to which the file belongs.
Group User	GU	Any user logged on to the group to which the file belongs, or who has been assigned that group as the home group.
Creating User	CR	The user who created the file. No other user can change the file access permissions.

At the account and group level, these permissions are assigned by the System Administrator, as described in more detail in *MPE V Accounting, Billing and Security* (32033-90136). At the file level, the file's creator can change the default permissions (Table 6-1) assigned by MPE. The procedures that follow describe how to find out what file access permissions apply to a file and how to change them at the file level.

Checking File Access Permissions

Although you have no control over the permissions at the account and group levels, you can find out which permissions apply with the LISTDIR5 utility. This is particularly helpful if you have trouble accessing a file in another account or group. To be able to look at a file in another account, that file must have READ access permission for ANY user.

To check file access permissions:

1. Enter `RUN LISTDIR5.PUB.SYS` to run the subsystem.

   The system displays the > (chevron) prompt.

2. Enter `LISTSEC,` followed by the file name, to display the file access permissions that apply to a file.

   ```
   >LISTSEC filename
   ```

   The system displays the file access permissions at the account, group, and user levels, similar to Example 6-4.

Example 6-4 shows how to display which file access permissions apply to which users for the file MYFILE. Note that file access permissions differ at the account, group, and file (user) level.

```
:RUN LISTDIR5.PUB.SYS

LISTDIR5 G.02.B0 (C)  HEWLETT-PACKARD CO., 1983
Enter 'HELP' FOR AID

>LISTSEC MYFILE

************************
FILE: MYFILE.MGR.PAT

SYSTEM       READ:              ANY
SECURITY     WRITE:             AC
  (ACCT)     APPEND:            AC
             LOCK:              ANY
             EXECUTE:           ANY

SYSTEM       READ:              ANY
SECURITY     WRITE:             AL, GU
  (GROUP)    APPEND:            AL, GU
             LOCK:              ANY
             EXECUTE:           AL, GU

SYSTEM       READ:              ANY
SECURITY     WRITE:             ANY
  (FILE)     APPEND:            ANY
             LOCK:              ANY
             EXECUTE:           ANY

FOR MGR.PAT:  READ, WRITE, APPEND, LOCK, EXECUTE
>
```

Example 6-4. Displaying File Access Permissions

Reassigning File Access Permissions

You have control over file access permissions at the file level for any files you created. Although you can add lockwords to keep people out of your files completely, you may want only to partially restrict access. To do so, you can change the file access permissions for your file from the default permissions listed in Table 6-1. For example, you may want to restrict the WRITE and APPEND permissions to allow other people to look at your file, but not to change it.

The :ALTSEC command lets you change the file access permissions assigned to your file at the file level. You can also temporarily remove the resulting restrictions with the :RELEASE command and reinstate them with the :SECURE command. For example, a colleague may need

to add information to a file with restrictions of the WRITE and APPEND permissions. By "releasing" the file, you temporarily remove these restrictions to allow this addition to your file. By "securing" the file, you reinstate the restrictions.

NOTE

Be careful when using the :RELEASE command. Your file is unprotected until you use the :SECURE command.

Restricting File Access Permissions

You can use the :ALTSEC command to change the file access permissions (READ, LOCK, APPEND, WRITE, or EXECUTE) of any disc files you create. (However, this does not affect the account and group security levels.) Also, if you defined a lockword, you must include the lockword in the :ALTSEC command.

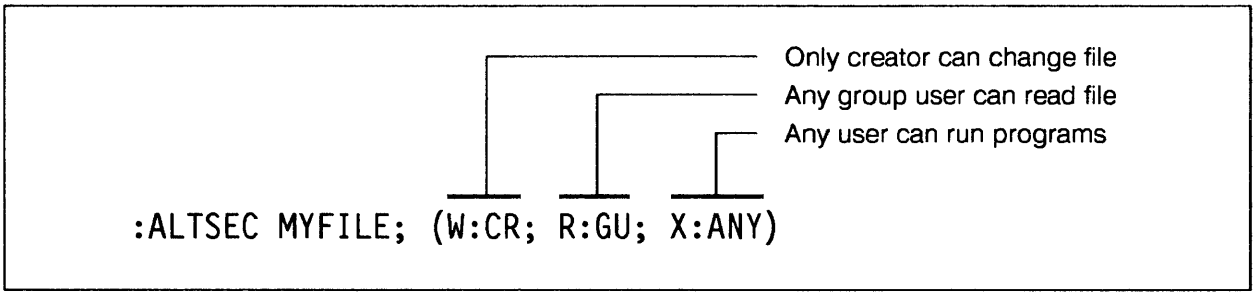
To restrict file access permissions:

Enter ALTSEC, followed by the file name, the access permission codes and user codes, and the lockword (if any).

You can include these codes in any order, but you must separate the access permission code (Table 6-1) from the user code (Table 6-2) with a colon. (Hint: think of the colon as an = sign). Follow each access/user permission with a semicolon.

```
:ALTSEC filename, (access code:user code; access code:user code; etc.)
```

Figure 6-8 shows how to assign the following file access permissions to the file MYFILE: WRITE access to the creator only; READ access to any group user; EXECUTE access to any user.



LG200077_049

Figure 6-8. Restricting File Access Permissions

Releasing a File

You may want to temporarily remove restrictions to file access permissions. For example, if other users want to copy your file, they must have the READ and WRITE access permission.

To temporarily remove restrictions to file access permissions:

Enter RELEASE, followed by the file name.

```
:RELEASE filename
```

As an option, you can also specify a file's user name and account name. This is necessary if you are working in another group when you want to release a file in your home group.

Securing a File

When other users no longer need access to your file, you can reassign the restrictions to file access permissions.

To reinstate restrictions to file access permissions:

Enter SECURE, followed by the file's name.

```
:SECURE filename
```

NOTE

Exception: Privileged files (like databases) cannot be released. You can see which files are privileged by entering LISTF, 2 at the colon prompt. Any files with negative file codes are privileged files.

Capabilities

Passwords prevent access to accounts, groups, and users by anyone who does not know the applicable password(s). File access permissions determine which users can access and alter specific files.

Capabilities further define the amount of control specific user categories have, thus adding another level of security. For example, very few users have the capability to make changes to the system. On the other hand, every user should be able to save files on disc. In this way, capabilities protect the system from unauthorized or unqualified users.

Capabilities are assigned at the account, user, and group level. Each capability is associated with a two-letter mnemonic, as shown in Table 6-3.

On each HP 3000, the account SYS and the user MANAGER are assigned the capability SM, as well as every other capability possible. This user, in turn, assigns capabilities to other accounts and users. Table 6-3 shows a complete list of capabilities. For details about the tasks associated with each capability, see *MPE V Accounting, Billing and Security* (32033-90136).

Table 6-3. Capabilities

Capability	Abbrev-iation	Account	Group	User
System Manager	SM	✓		✓
System Supervisor	OP	✓		✓
Account Manager	AM	✓		✓
Account Librarian	AL	✓		✓
Batch Access	BA	✓	✓	✓
Use Communications Software	CS	✓		✓
Diagnostician Attribute	DI	✓		✓
Extra Data Segments	DS	✓	✓	✓
Group Librarian	GL	✓		✓
Interactive Access	IA	✓	✓	✓
Multiple RIN	MR	✓	✓	✓
Network Administrator	NA	✓		✓
Node Manager	NM	✓		✓
Use Nonsharable Devices	ND	✓		✓
Use Private Disc Volumes	UV	✓		✓
Privileged Mode	PM	✓	✓	✓
Process Handling	PH	✓	✓	✓
Programmatic Sessions	PS	✓		✓
Save User Files Permanently	SF	✓		✓
Use User Logging Facility	LG	✓		✓
Create Volume Sets	CV	✓		✓

LG200027_013

General User Capabilities

Capabilities assigned to individual users depend on that user's responsibilities. Unless specified otherwise, MPE automatically assigns the capabilities summarized in Table 6-4. These are the basic capabilities necessary to do useful work on the HP 3000.

Table 6-4. Default User Capabilities

Capability	Code	Allows user to:
Interactive Access	IA	Initiate and maintain sessions.
Batch Access	BA	Run batch jobs.
Non-Sharable	ND	Use printers, tape drives, and other devices that must be accessed in turn.
Save File	SF	Save files permanently on disc.

Checking Capabilities

Users with AM (Account Manager) or SM (System Manager) capability can get extensive information about the account structure, including passwords and lockwords.

As a general user, you can use the LISTDIR5 utility to find out account information, including the capabilities assigned to you and your account. This may be useful if you are having problems with certain commands or operations.

To display user capabilities

1. Enter `RUN LISTDIR5.PUB.SYS.`

   The system displays the chevron (>) prompt.

2. Enter `LISTUSER`

   The terminal displays a screen similar to Example 6-5.

Example 6-5 shows the information displayed by the LISTDIR5 subsystem. The section named "CAP" shows the capabilities (AM, AL, GL, ND, SF, IA, BA) of this sample user. This user, for example, has the AM (Account Manager), AL (Account Librarian), and GL (Group Librarian) capabilities in addition to the default capabilities listed in Table 6-4.

Note also that the password is not displayed. If the System Manager were to use the same command (with the ;PASS parameter), the additional capabilities assigned to the SM user would also display the password.

```
:RUN LISTDIR5.PUB.SYS

LISTDIR5.G.02.BO (C) HEWLETT-PACKARD CO., 1983
Enter 'HELP' FOR AID

>LISTUSER
**********************
USER: MGR.PAT

HOME GROUP: PUB                 PASSWORD:  **
MAX PRI:150                     LOC ATTR: %0
LOGON CNT: 2
CAP:  AM, AL, GL, ND, SF, IA, BA

LOGON GROUP:  PUB     SESSION # S407
LOGON DEV #:  52
```

Example 6-5. Displaying User Capabilities

7

Working with Files

Chapter Overview

MPE's "file system" controls the processing of all files, including storage and transfer between various devices. This chapter introduces you to MPE's file system and associated concepts, including:

- Introduction
- Basic file handling tasks
  - Displaying information about files
  - Finding files
  - Displaying a file's contents
  - Renaming files
  - Deleting files
  - Copying files
- Advanced file handling tasks
  - Overview of the file system
  - File characteristics
  - Creating files with the `:BUILD` command
  - Device files
  - Writing file equations
  - Checking and deleting file equations
  - Printing files
  - Displaying additional information about files
  - Saving temporary files

Introduction

To the computer, a file is a basic unit that holds information to be processed. Once organized into a file with a name, the computer can find this information and process it according to the user's instructions. In this way, a computer file is similar to a file folder in that it contains information stored in a single, named location. And like file folders, computer files make it easy to find, manipulate, and transport the information they contain.

Most files are stored on the disc drives attached to the computer. When MPE needs to process a file, it first "opens" that file. It then "reads" the file, which means it copies it into its internal memory. It can also "write to" the file, which means it changes the file contents during processing. When the file is no longer needed, the computer "closes" the file, which means it copies the altered file back to disc, where it is stored.

MPE can also send files from one device to another. For example, files stored on disc are frequently sent to a terminal to be displayed or a printer to be printed.

Although only this chapter specifically discusses files and their characteristics, most of the other chapters also deal implicitly with files. This is because almost all the computer's work involves manipulating files. In particular, Chapter 5, "Subsystems and Utilities", discusses how MPE subsystems allow you to work with files to accomplish the tasks shown in Table 7-1.

For additional information about files, refer to the *MPE V File System Reference Manual* (30000-90236).

Table 7-1. Subsystems and Files

Subsystem	Function
EDITOR	Creates, edits, and prints text files.
SORT	Sorts files according to selected criteria.
MERGE	Merges any sorted files.
STORE	Copies files to tape for backup and transfer.
RESTORE	Copies files copied to tape with the `:STORE` command back to disc.
FCOPY	Copies files between groups and accounts on same system, copies files from one medium to another, compares file contents, and more.
SPOOK	Allows access to spoolfiles.

This chapter consists of two parts. "Basic File Handling" tells you how to use commands to do some basic things to previously created files, without using the subsystems or applications that created the file. "Advanced File Handling" explores in more detail how files are constructed and processed. This gives you additional control over the system, similar to the control provided by the optional parameters in MPE commands.

Summary

Most applications and subsystems have already established a common way to deal with files. And, like commands without optional parameters, they work just fine in most cases. However, MPE's file system options provide additional flexibility and control. For example, you can control a file's physical characteristics, such as its size, the blocking factor, the disc allocation, whether it is permanent or temporary, and so on. You can also determine to which devices the file is sent. This information helps you to customize MPE.

NOTE

This chapter includes step-by-step instructions for several tasks. To avoid repetition, assume the following:

1. At each step, you type information at the colon (:) prompt, unless specified otherwise.

2. After entering information, you press Return.

Basic Uses of the File System

Files created by users are called "user-defined files." For example, if you write a memo using the EDITOR subsystem and save it with the /KEEP subcommand, that memo becomes a file. Besides text, files can also contain other information, such as bit maps of graphic images, or "executable code", which is a program ready to be run.

Because user-defined files are usually stored on discs until they are deleted (purged), they are more commonly called "disc files." These files have an "address", which is similar to a person's address in that it tells the system where to find that file. This address is listed in the MPE directory. Disc files can be used by any users with the necessary security provisions. (Refer to Chapter 6, "Basic Account Structure", for a discussion of lockwords and file access permissions.)

This section describes how to use MPE's file commands to manipulate disc files. This includes displaying files, finding specific files, renaming files, and deleting files.

Displaying Information about Files

The :LISTF command, in its simplest form, is one of MPE's most commonly used commands. It shows what disc files exist on the system. It is also useful to check that files you tried to copy or delete were indeed copied or deleted.

Used by itself, the :LISTF command displays the files available to your logon group and account. To find out what files exist in other groups and accounts, you must specify the desired group or account.

When used with parameters, the :LISTF command also displays various levels of detail about any or all files. This is described in more detail in "Advanced File Handling."

Example 7-1 shows how to display a directory of files in the user's logon group and account, as well as a listing of files outside that group and account. Note that the files are listed in alphabetical order.

```
:LISTF

FILENAME

ALLFILE   BOOKS     CONTS     EMP       ERRORS    FILE1
FILE2     FILE3     GATES     MYFILE    MYUDC     SAMPLE
STREAMIT  TEST      TONY

:LISTF RECORDS.ACCT

FILENAME

ADDALL    BKPROC    CUST1     CUST2     CUST3     CUST4     DELNEW
ENDWK     ENDMO     FORGO     FILESET   IRS       PROC1     PROC2
TALLY     TERM      TOTALS    XEC       XECCALC   XECSUB
```

Example 7-1. Sample :LISTF Display

Finding Files

Finding a file in the sample display in Example 7-1 would be easy, even if you are not sure of its exact name. However, at times you may need a file in an account that has hundreds of files. To make it easier to find that file, you can narrow down the display of files by including "wild card" characters in the :LISTF command.

A wild card character can stand for any letter, number, or symbol within the file name. For example, you could search for files that contain a number, or those that begin or end with a certain letter. You can also use wild card characters to specify all groups or accounts.

MPE uses the following wild card characters:

@	Character strings of any length
?	A single alphanumeric character
#	A single numeric character

Example 7-2 shows how to find a file in the user's logon group and account that starts with the letter "S." It also shows how to display all files in all accounts that contain the characters "UDC", as well all files in a sample PUB group and SYS account that contain a numeric character.

```
:LISTF S@

FILENAME

SAMPLE     STREAMIT


:LISTF @UDC.@.@

FILENAME

MYUDC      UDC        ALLUDC     UDCFILE    CKUDC      GAMEUDC
LOGUDC     WPUDC      JOBUDC     UDCLOG


:LISTF @#.PUB.SYS

FILENAME

INQUI1     OUT2       OUT3       N22IRS
```

Example 7-2. Using Wild Card Characters to Find Files

Displaying a File's Contents

To display a file, use the appropriate command(s) in the program or subsystem that created the file. For example, to display the contents of an EDITOR file, you would use the /TEXT command to retrieve the file from disc and the /LIST ALL command to display the file on the terminal.

Renaming Files

The :RENAME command lets you change a file's name. It can also add or change lockwords to keep other people from opening the file. You may also want to change a file's name when its contents change, to keep the file name descriptive of the contents.

NOTE

Only the file's creator can rename that file. Also, you cannot rename a file while someone is using it. If you try, you will get the following message:

```
EXCLUSIVE VIOLATION:  FILE BEING ACCESSED (FSERR 90)
OPEN FAILED ON FILE, NOT RENAMED  (CIERR 372)
```

To rename a file:

1. Enter RENAME, followed by the file name and the new file name.

   ```
   :RENAME filename, new filename
   ```

2. (Optional) Enter LISTF to check the result.

To rename a file and add a lockword:

1. Enter RENAME, followed by the file name, the new file name, a slash, and the lockword.

   ```
   :RENAME filename, new filename/lockword
   ```

2. (Optional) Enter LISTF to check the result.

Example 7-3 shows how to rename the file BOOKS (Example 7-1) to BOOKS2 and add the lockword BOZO. Note that the lockword does not show in response to the :LISTF command.

```
:RENAME BOOKS, BOOKS2/BOZO
:

:LISTF

FILENAME

ALLFILE          BOOKS2     CONTS      EMP        ERRORS     FILE1
FILE2            FILE3      GATES      MYFILE     MYUDC      SAMPLE
STREAMIT         TEST       TONY
:
```

Example 7-3. Renaming a File

Deleting (Purging) Files

Although the disc drives connected to the HP 3000 can hold vast amounts of information, there are limits. Since each file takes up space on disc, it is necessary to get rid of old files periodically. It is a good habit to delete files you no longer need on a regular basis with the :PURGE command.

Once a file is deleted, the only way to retrieve it is from the system backup tape. Since that procedure is somewhat complicated, be careful about which files you delete. You know the computer has purged the file when the colon (:) prompt appears in response to the :PURGE command. If you now enter the :LISTF command, that file name no longer appears in the list of files.

As a general user you can only delete files for which you have the WRITE capability. Do not delete any files that you did not create, except under special circumstances.

To delete a file from disc:

Enter PURGE, followed by the file name of the file you want to delete.

```
:PURGE filename
```

2. (Optional) Enter LISTF to check the result.

Example 7-4 shows how to delete the file GATES from the files shown in Example 7-1. The colon prompt means that MPE executed the command and is ready for the next command.

```
:PURGE GATES
:

:LISTF

FILENAME

ALLFILE          BOOKS2     CONTS      EMP        ERRORS     FILE1
FILE2            FILE3      MYFILE     MYUDC      SAMPLE     STREAMIT
TEST             TONY
:
```

Example 7-4. Deleting (Purging) a File

NOTE

Do not use the :PURGE command to delete KSAM files or database files. Instead, use the utilities provided by those programs.

Copying Files

To copy files, use the FCOPY subsystem, as described in Chapter 5, "Utilities and Subsystems". To copy between systems, use the DSCOPY subsystem. For procedures, see Appendix B.

Other Basic File Handling Tasks

There are several other basic file handling tasks. This includes editing files, printing files, backing up files to tape, sorting the contents of files, and so on. In most cases, you will do these tasks with a program or subsystem specifically designed for that purpose. For detailed procedures, refer to Chapter 5, "Utilities and Subsystems".

In addition, you must be aware of security features that affect which files you can look at and manipulate. These features are discussed in Chapter 6, "Basic Account Structure".

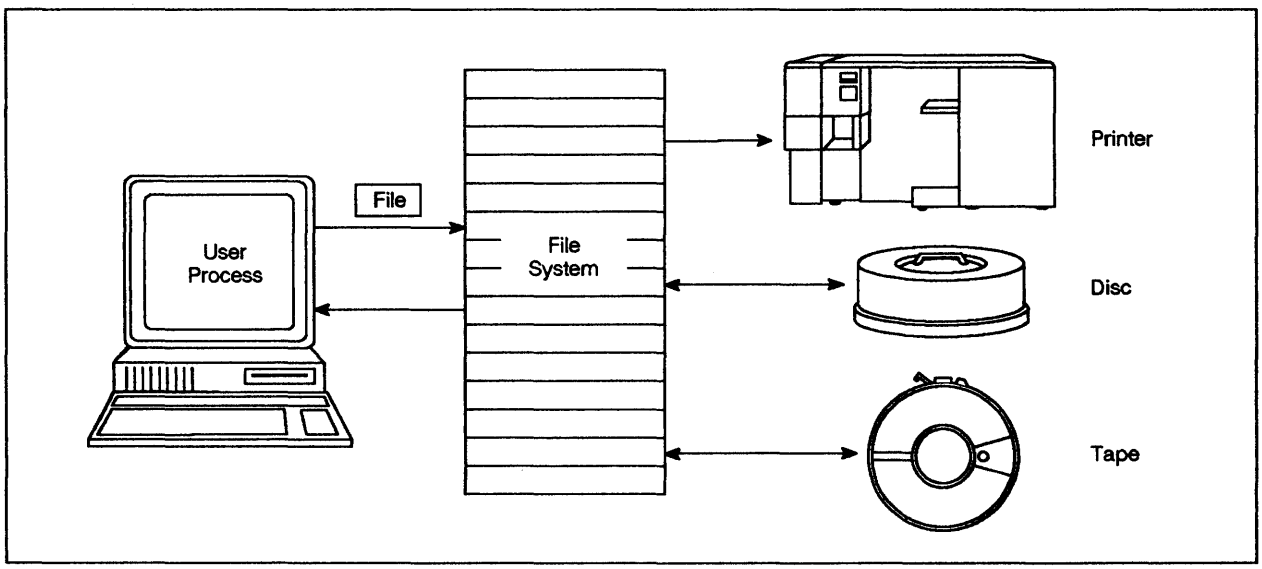
Advanced Uses of the File System

So far, this chapter has described how to use basic MPE commands to display available files, to rename files, and to delete files. This section examines MPE's file system more closely, including an overview of the file system and procedures for controlling more directly the creation and transfer of files.

File System Overview

MPE's file system manages all files being stored on disc, transferred between user processes, and transferred between processes and peripheral devices. All MPE file operations are performed through a set of system-supplied procedures called "intrinsics," which are fully described in the *MPE V Intrinsics Reference Manual* (32033-90007).

Figure 7-1 shows an overview of this process.



LG200077_050

Figure 7-1. Overview of the File System

Creating Files

Most subsystems and applications automatically create files with the appropriate characteristics, which determine how a file's contents are stored on disc. Most general users are not concerned about these characteristics, since the defaults are usually appropriate for the task at hand.

File Characteristics

However, once you become an experienced user, you may be concerned about how to make programs run most efficiently by defining a file's characteristics more precisely. Although you

cannot alter the physical characteristics of existing files, you can create a new file with the desired characteristics and copy the contents of the old file into the new file.

For descriptions of file characteristics, refer to "Record Structure and Blocking" in the *MPE V File System Reference Manual* (30000-90236). You can find out what file characteristics apply to a specific file by using the :LISTF command with the command's additional parameters. For procedures, refer to "Displaying File Characteristics" in this chapter.

File characteristics include:

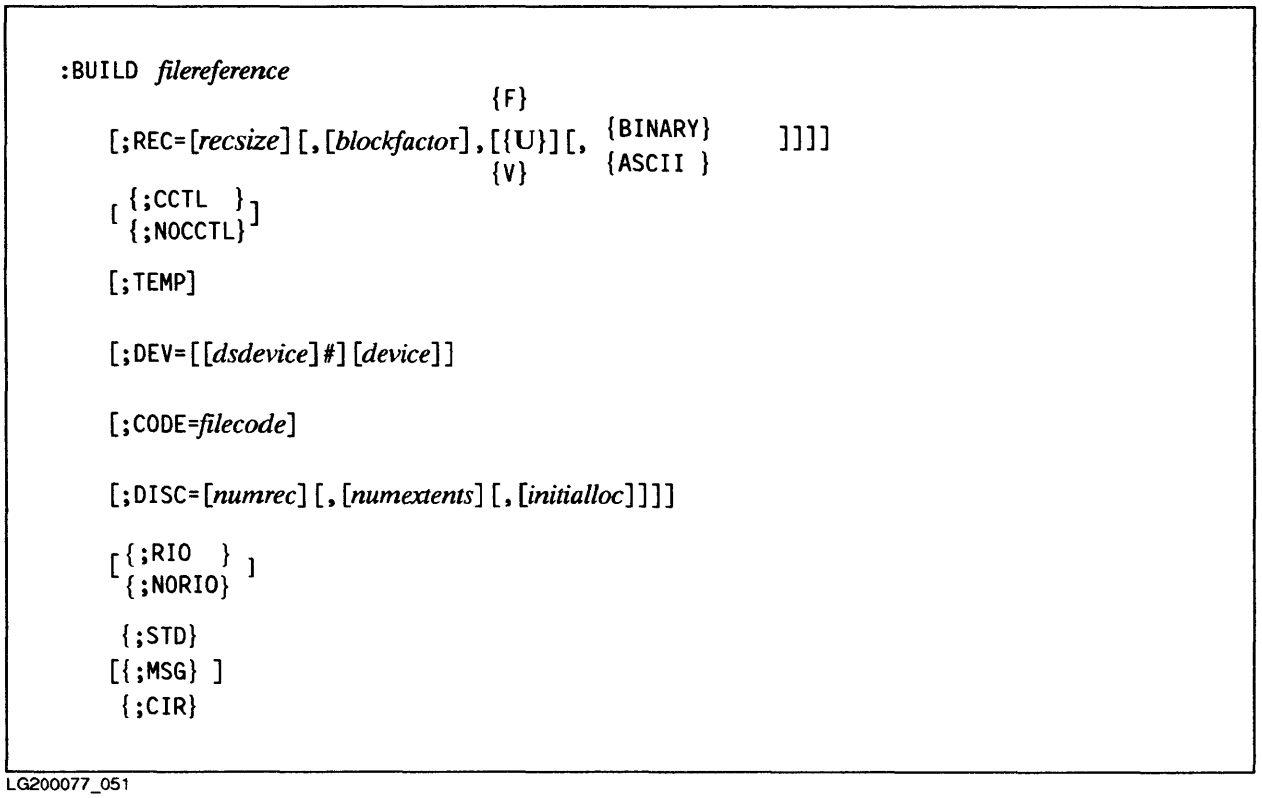
- Record size
- Record type
- The blocking factor
- Whether information is stored in ASCII or BINARY code
- Whether or not to include carriage control characters
- Whether the file is temporary or permanent
- What devices are associated with the file
- The file code
- Information about number of extents allocated
- Special ways of accessing the file

The :BUILD Command

The :BUILD command builds a new file and immediately allocates space for it. By using its optional parameters, you can also specify the file characteristics.

By default, :BUILD creates a permanent file with fixed length records of 128 words each, a blocking factor of 1, coded in binary code. The record limit is 1023, with a maximum of eight extents, with one extent allocated initially. The *MPE V Commands Reference Manual* (32033-90006) describes each of these characteristics in detail.

Figure 7-2 shows the :BUILD command's syntax.



LG200077_051

Figure 7-2. Syntax of the :BUILD Command

To create a file using the :BUILD command:

Enter BUILD, followed the file name and any optional parameters to specify desired characteristics.

```
:BUILD filename; characteristics
```

The system allocates space for that file and assigns it the specified characteristics.

Example 7-5 illustrates how to create a file called ENROLL with a record size of 80, a blocking factor of 3, with fixed-length (parameter was omitted, therefore MPE assigned the default), ASCII-coded records. Disc space is specified for a maximum number of 500 records in four extents, with one extent allocated initially.

```
:BUILD ENROLL;REC=-80,3,,ASCII;DISC=500,4,1
:
```

Example 7-5. Creating a File with the :BUILD Command

Device Files

Files created by users are permanently stored on disc, unless the user specified otherwise. Called "disc files," these files have the characteristics described in the previous section. All users with the necessary security provisions can access disc files.

In contrast to disc files, "device files" are not tangible entities. Instead, they are a logical link that allows you to treat peripheral devices as though they were files. This makes it possible for users to access devices and disc files in a standard way. As a result, the file is device independent, which means it is not restricted to the same device every time a program is run.

System-defined Files

Whenever you initiate a job or session, MPE automatically sets up two device files: $STDIN and $STDLIST. Because they are created by MPE, they are called "system defined files." MPE also creates other system defined files that temporarily hold data generated during program execution. You can recognize system defined files by the dollar ($) character. The characteristics associated with system defined files cannot be changed.

Table 7-2 summarizes the system defined files.

Table 7-2. System Defined Files

Filename	Specifies
$STDIN	The device from which MPE expects input. The default is the terminal for both sessions and jobs.
$STDLIST	The device to which MPE sends output. The default is the terminal for sessions or the line printer for jobs.
$NEWPASS	A temporary file created automatically to which newly generated information is written.
$OLDPASS	A temporary file created when the file $NEWPASS is closed.
$NULL	An empty file if used as input. A bottomless hole when used as output.

The $STDIN and $STDLIST System-defined Files

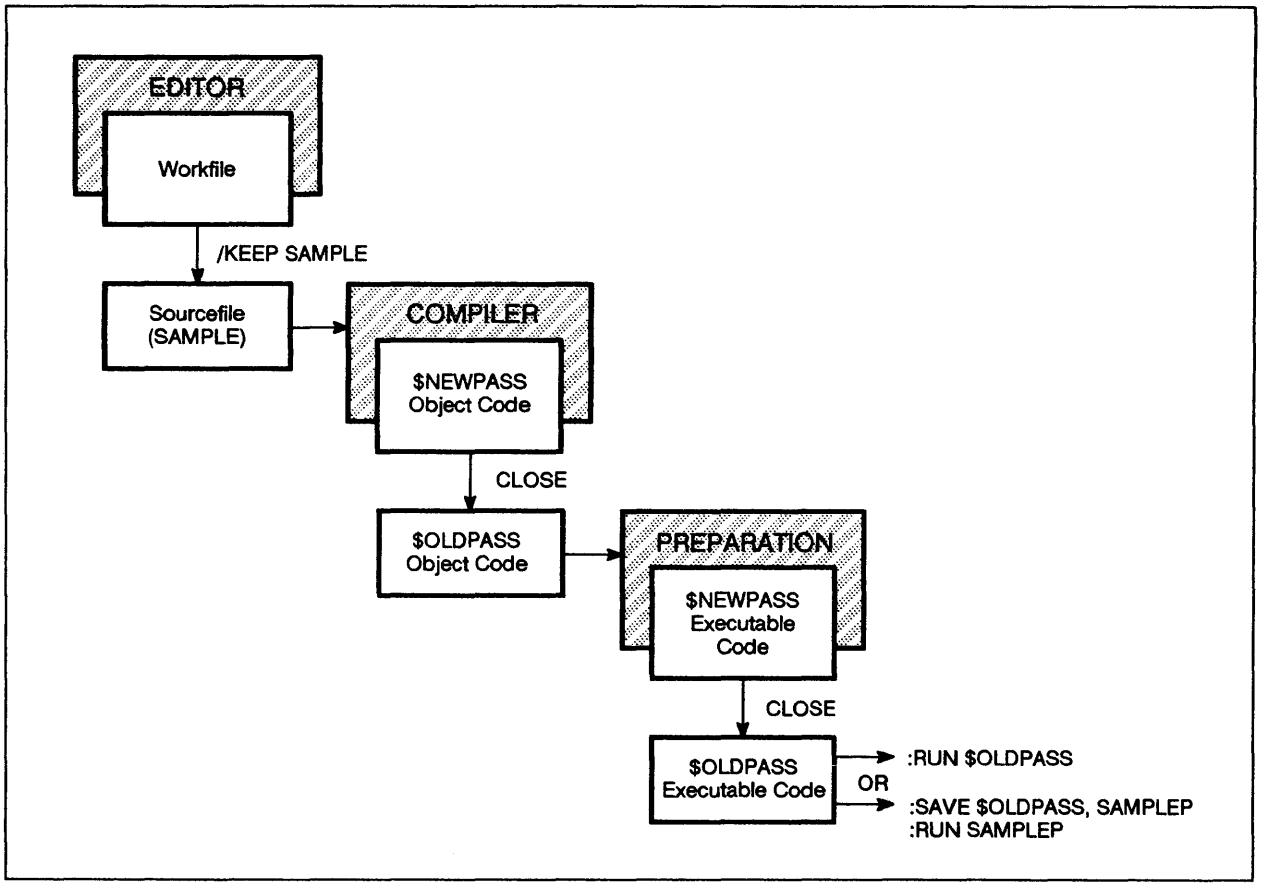
When you log on, MPE automatically creates a system-defined file called $STDIN, which corresponds to the device from which it expects input. It also creates a file called $STDLIST, to which it sends output. As you can see in Table 7-2, the actual devices that correspond to these files depend on whether you are in a session or job. For a session, the input device ($STDIN) and output device ($STDLIST) are the same: the terminal. This is because the terminal is used for both input and output. For example, when you enter the :LISTF command on the keyboard (input), the resulting listing (output) is displayed on the screen.

The $NEWPASS, $OLDPASS, and $NULL System-defined Files

In addition to system-defined files that define the devices automatically associated with input and output files, MPE creates temporary files that hold data generated during processing. In this way, output generated by each process can be passed to the next process as input. When the program is completed, you can use the final output file as necessary.

For example, the different steps in compiling a program create several $NEWPASS (output) files. $NEWPASS files automatically change to $OLDPASS (input) files when the files close and are passed to the next process. When the final process is complete, you can run the program with the command :RUN $OLDPASS or save the $OLDPASS file under a different name and then run the program. Figure 7-3 illustrates this process.

The final MPE system defined file is "$NULL." You can send output that you want to delete to this file. This is handy for testing programs when you do not want to keep the output.



LG200077_052

Figure 7-3. The $NEWPASS and $OLDPASS System-defined Files

Subsystem Formal File Designators

The system defined device files $STDIN and $STDLIST are always available to automatically route input and output. In addition, MPE subsystems automatically route files according to predefined device files associated with the formal file designators shown in Table 7-3. In most cases, this is the line printer, which is designated "LP." For example, when MPE encounters the formal file designator MAILPRNT, it automatically sends the file to the line printer when you enter the `/PRINT` command.

By writing your own file equations, you can reroute files created by subsystems. For example, the file equation `:FILE MAILPRNT;DEV=PP` would send the file to the laser printer (PP) instead of the line printer (LP) when you enter the `/PRINT` command. In other words, you can let the subsystem decide where a file is sent, or you can decide for yourself.

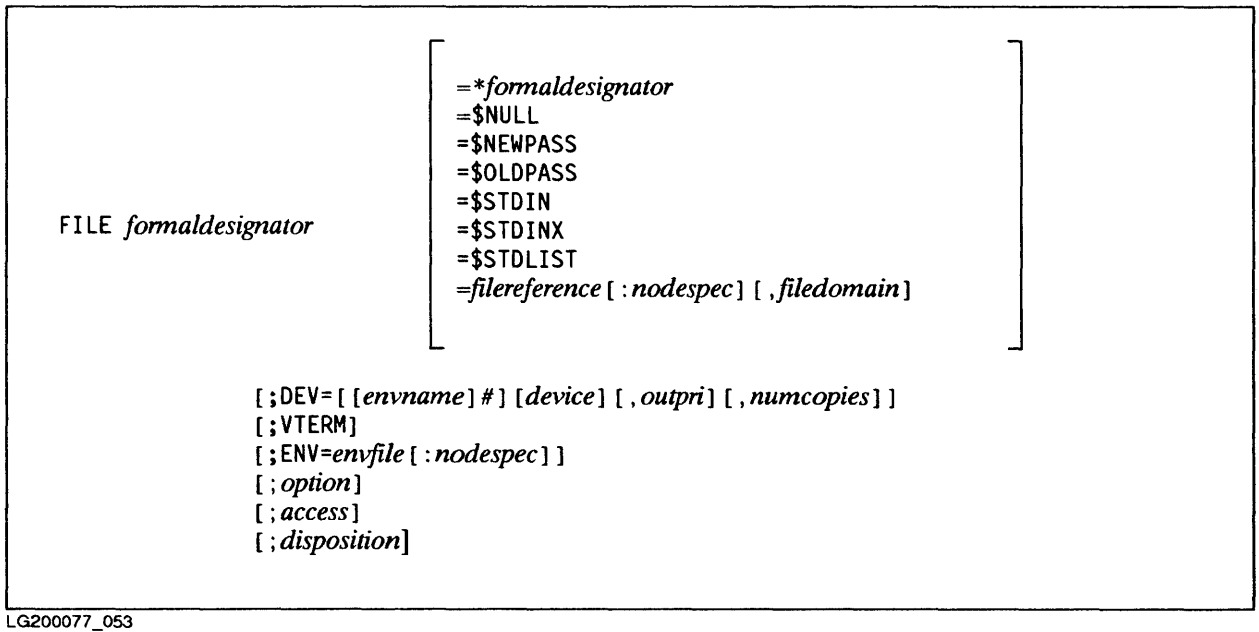
For a complete list of the formal file designators associated with various subsystems, see Appendix B in the *MPE V Commands Reference Manual* (32033-90006).

Table 7-3. Sample Formal File Designators

Program	Formal File Designator	Command
EDITOR	EDTLIST	LIST ALL, OFFLINE
HPDESK	MAILPRNT	PRINT
HPLIST	LISTOP	PRINT
HPSLATE	SLLIST	OFFLINE PRINT
QUERY	QSLIST	OUT=LP
TDP	SLP	FINAL
STORE	SYSLIST	STORE ...;SHOW
RESTORE	SYSLIST	RESOTRE ...; SHOW

Writing File Equations

The :FILE command, which is shown in Figure 7-4 and described in detail in the *MPE V Commands Reference Manual* (32033-90006), is one of MPE's most versatile, powerful, and complex commands.



LG200077_053

Figure 7-4. The :FILE Command

Using the :FILE command is commonly called "writing a file equation." This section describes the three most common uses for writing file equations:

- To link actual input and output files with generic file references within programs.
- To create device files to reroute output files.
- To link disc files with device files by "backreferencing."

File equations remain in effect until you use the :RESET command, rewrite the file equation, or log off. They are also lost if the system fails.

Linking Actual Files with Generic File References

Many programs require input files that change periodically. Since it would be impractical to repeatedly rewrite the program to reflect the names of the new input files, it is a common practice to substitute a generic file name (formal file designator) in the program's code whenever a particular input file is required. Unlike the actual input files, the formal file designator remains constant. In this way, the program's code does not need to be changed.

For example, assume that a program (called DAILYINV) calculates the daily total of invoices received. To do so, it requires a new file each day that contains that day's figures. The files for each day are named by date, such as JAN12 and JAN13. The program itself, however, simply calls those files TOTALS whenever it refers to them in its code. Therefore, when the program is run, there must be a way to link the generic name TOTALS to the actual file names JAN12 or JAN13.

In the same way, the actual output file also changes each time the program is run. As with the input files, there must be a way to link the program's formal file designator with the actual output file that holds the results of the day's run. MPE's file equations provide that link.

To write a file equation to link formal and actual file designators:

Type FILE, followed by the formal file designator, an equal sign, and the actual file designator.

```
:FILE formal file designator=actual file designator
```

When the program executes, it reads from and writes to the actual files associated with the formal file designators.

Example 7-6 shows how to write file equations for the program DAILYINV. These file equations link the program's formal file designator TOTALS with the actual file input file JAN12, and the formal file designator REPORT with the actual output file REP12.

```
:FILE TOTALS=JAN12

:FILE REPORT=REP12
:
```

Example 7-6. Writing File Equations to Link Formal File Designators with Actual File Designators

Creating Device Files

In Example 7-6 , the output file REP12 is sent to disc by default. However, you can send it to another device by writing a file equation that links the output file with that device. For example, the file equation `:FILE REPORT;DEV=LP` would send the file to the line printer instead.

In addition, you can create device files that link a formal file designator not associated with an actual input or output file to device specifications. These specifications include the device, as well as the characteristics associated with that device. This includes the number of copies to be printed, the output priority, and the "environment" for laser printers, which contains such specifications as the page size, the font, and other printing requirements.

To create a device file with a file equation:

Enter `FILE`, followed by the formal file designator for a device and any device specifications.

```
:FILE formal file designator;DEV=specifications
```

Example 7-7 shows how to write a file equation that creates a device file "PP," which is associated with the page printer and the printing requirements associated with the environment `pica.hpenv.sys`. (These options, of course, are optional parameters of the `:FILE` command. For a full description of these options, refer to the *MPE V Commands Reference Manual* (32033-90006).

```
:FILE PP;DEV=PP;ENV=PICA.HPENV.SYS
:
```

Example 7-7. Creating Device Files to Reroute Output Files

Backreferencing File Equations

Once you have create a device file such as in Example 7-7, you can easily link that device file with the formal file designator assigned by a program or an MPE subsystem. For example, you can backreference any of the formal file designators in Table 7-3 to a device file you created.

To backreference a file equation:

Enter FILE, followed by the formal file designator, a = sign, an asterisk, and the name of the device file.

```
:FILE formal file designator=*filename
```

Example 7-8 shows how to link the device file created in Example 7-7 with backreferencing. It links this device file with the formal file designator of the DAILYINV program, as well the as those of files created by the EDITOR and HPDESK subsystems. It also shows how backreferencing can be used with the FCOPY subsystem.

Example 7-9 shows how to achieve the same result without creating a device file and using backreferencing. As you can see, creating a device file and backreferencing can save considerable typing.

```
:FILE PP;DEV=PP;ENV=PICA.HPENV.SYS
:FILE REPORT=*PP
:FILE EDTLIST=*PP
:FILE MAILPRNT=*PP
:FOCPY FROM=MYFILE;TO=*PP
:
```

Example 7-8. Backreferencing to a Device File

```
:FILE REPORT=DEV=PP;ENV=PICA.HPENV.SYS
:FILE EDTLIST=DEV=PP;ENV=PICA.HPENV.SYS
:FILE MAILPRNT=DEV=PP;ENV=PICA.HPENV.SYS
:FOCPY FROM=MYFILE;TO=DEV=PP;ENV=PICA.HPENV.SYS
:
```

Example 7-9. Comparison to Example 7-8

Summary: Writing File Equations

File equations give you considerable flexibility by giving you the following capabilities:

- Linking actual files with generic file references within programs.
- Creating device files to reroute output to devices other than those specified by the system defined files $STDLIST, or the devices associated with the formal file designators of various subsystems.
- Using backreferencing to link device files created with file equations to actual files created by programs or subsystems.

For additional capabilities, consult the *MPE V Commands Reference Manual* (32033-90006).

Checking Existing File Equations

The :LISTEQ command displays the device files created in current session with file equations. This command is useful if you have set up a number of file equations and need to check where you sent which files.

To check file equations set to the current session:

Enter LISTEQ.

The system displays the current file equations, as shown in Example 7-10.

```
:LISTEQ
FILE TOTALS=JAN12
FILE REPORT=REP12
FILE PP;DEV=PP;ENV=PICA.HPENV.SYS
FILE REPORT=*PP
FILE EDTLIST=*PP
FILE MAILPRNT=*PP
FOCPY FROM=MYFILE;TO=*PP
:
```

Example 7-10. Displaying Existing File Equations

Clearing File Equations

You can clear any file equations set up by the user for the current session with the :RESET command.

To clear a file equation:

Enter RESET, followed by the formal file designator.

The system deletes the specified file equation.

To clear all file equations:

Enter RESET, followed by the @ character.

The system deletes all file equations. However, when you log off, file equations set up by the System Administrator will again be active the next time you log on.

Example 7-11 shows how to delete and check the deletion of the first file equation listed in Example 7-10.

```
:RESET TOTALS

:LISTEQ

FILE REPORT=REP12

FILE PP;DEV=PP;ENV=PICA.HPENV.SYS

FILE REPORT=*PP

FILE EDTLIST=*PP

FILE MAILPRNT=*PP

FCOPY FROM=MYFILE;TO=*PP
:
```

Example 7-11. Deleting File Equations

Printing Files

You can use the FCOPY subsystem, as described in Chapter 5, or the appropriate printing commands in the program that created the file, as shown in Table 7-3. For example, to print a file created with the HPDESK program, you would use the PRINT command.

Displaying File Characteristics

You can display file characteristics by using the :LISTF command's optional parameters or the LISTDIR5 subsystem. When entered without parameters, the :LISTF command displays a list of permanent files, as well as any temporary files created during the current session. A typical :LISTF display is shown in Example 7-1.

Displaying File Characteristics with the :LISTF Command

You can display various levels of detail about any or all files by specifying the :LISTF command's optional parameters. These parameters display additional information about the file(s) specified, including the options defined either by default or by the :BUILD or :FILE commands. This may be helpful if you want to create a new file with characteristics similar to an old file.

Table 7-4 shows those options of the :LISTF command accessible to the general user. Users with special capabilities have additional options, as described in the *MPE V Commands Reference Manual* (32033-90006).

Table 7-4. :LISTF Display Options

Command	Resulting Display
LISTF	Displays all file names in user's account.
LISTF, 1	Displays file names, file codes, record size, current end-of-file location, and maximum number of records allowed.
LISTF, 2	Displays the same information as the 1 parameter, as well as the blocking factor, the number of disc sectors in use, the number of extents currently allocated, and the maximum number of extents allowed.

To display a file's code, record size, end-of-file location, and maximum number of records:

Enter LISTF, the file name, and the parameter 1

```
:LISTF filename, 1
```

The system displays a screen similar to Example 7-12.

Example 7-12 shows how to find out the record size of the file SAMPLE, which is one of the pieces of information displayed by the parameter 1.

```
:LISTF SAMPLE, 1

ACCOUNT= PAT        GROUP =     PUB

FILENAME  CODE   ---------LOGICAL RECORD -------

                 SIZE  TYP      EOF         LIMIT

SAMPLE            80B   FA       7            7
:
```

Example 7-12. :LISTF Display with Parameter 1

Displaying File Characteristics with the LISTDIR5 Utility

You can also use the LISTDIR5 subsystem to display additional information about files, including a history of the file and the levels of security that apply to it.

To display a file's characteristics:

1. Enter RUN LISTDIR5.PUB.SYS to run the subsystem.

   The system displays the > prompt

2. Enter LISTF, followed by the file name, to display the file's characteristics.

   ```
   :LISTF filename
   ```

Example 7-13 shows how to display the file characteristics for a file named SAMPLE.

```
:RUN LISTDIR5.PUB.SYS
>

>:LISTF SAMPLE
******************************
FILE: SAMPLE.PUB.PAT


FCODE:   0                      FOPTIONS:  STD,BINARY,FIXED
BLK FACTOR:  1                  CREATOR:  **
REC SIZE:  256(B)               LOCKWORD: **
BLK SIZE:  128(W)               SECURITY-READ:  ANY
EXT SIZE:  128(S)                         WRITE:                ANY
# REC:  0                                 APPEND:               ANY
# SEC:  128                               LOCK:                 ANY
# EXT:  1                                 EXECUTE:              ANY
MAX REC:  1023                            **SECURITY IS ON
MAX EXT:  8                     COLD LOAD ID: %22666
# LABELS:  0                    CREATED:   THU, 14 JAN 1988
MAX LABELS:  0                  MODIFIED:  THU, 14 JAN 1988
                                           3:31 PM
DISC DEV  #:  3                 ACCESSED:  THU, 14 JAN 1988
DISC TYPE:  3                   LABEL ADR:  **
DISC SUBTYPE:  8                SEC OFFSET: %1
CLASS:  DISC                    FLAGS:  NO ACCESSORS
FCB VECTOR:  %0                 %0
```

Example 7-13. Using the LISTDIR5 Utility

Displaying Information about Temporary Files

As you know, creating temporary files is useful if you will not need the information again after you log off. In that way, you do not have to remember to purge those files.

The :LISTFTEMP command lets you find out which temporary files exist. This lets you decide whether you want to make any temporary files permanent before you log off. The parameters that work with the :LISTF command to display file characteristics also work with the :LISTFTEMP command.

To list temporary files:

Enter LISTFTEMP

In response, MPE displays all temporary files in the user's account, as shown in Example 7-14.

```
:LISTFTEMP

TEMPORARY FILES FOR PAT.MGR

$OLDPASS
K0281420
APPB
:
```

Example 7-14. Displaying Temporary Files

Changing Temporary Files to Permanent Files

You can "save" temporary files created by the TEMP parameter in the :BUILD command, as well as system-created $OLDPASS files that contain data generated during processing. Unless you make temporary files permanent with the :SAVE command, those files will be lost at the end of the current job or session.

To "save" a user defined temporary file:

Enter SAVE, followed by the file name.

```
:SAVE filename
```

The system saves the file on disc and assigns it an address in the directory. All users can now access this file (security provisions permitting).

To "save" the system-defined file $OLDPASS:

Enter SAVE, followed by $OLDPASS and the new permanent file name.

```
:SAVE $OLDPASS, filename
```

The system saves the file on disc and assigns it an address in the directory. All users can now access this file (security provisions permitting).

Example 7-15 shows how to make the file $OLDPASS, created by the compiler, permanent as a program file called REPORTS.

```
:SAVE $OLDPASS, REPORTS
:
```

Example 7-15. Saving a Temporary File

8

Working in Sessions

Chapter Overview

This chapter introduces the necessary concepts, commands, and procedures to efficiently work in interactive sessions on the HP 3000.

This chapter includes the following topics:

- Introduction
- Starting and stopping sessions
- Correcting and repeating commands
- Running programs and subsystems
- Interrupting and aborting commands, programs, and subsystems
- Monitoring the session
- Sending and receiving messages
- Session security

Introduction

There are two basic ways of working with the HP 3000, "sessions" and "jobs."

In a session, you enter commands at the terminal keyboard one at a time. The computer's response to each command is displayed on the terminal screen. This process, which resembles a conversation, is known as "interactive processing." In a job, you send commands to the computer to be processed as a single package. This process is also known as "batch processing."

For an illustration of this concept, refer to Figure 1-7 in Chapter 1, "Basic Concepts."

NOTE

This chapter includes step-by-step instructions for several tasks. To avoid repetition, assume the following:

1. At each step, you enter information at the colon (:) prompt, unless specified otherwise.
2. After entering information, you press Return.

Starting and Stopping Sessions

This section describes the :HELLO command, which initiates an interactive session between the user's terminal and the computer, and the :BYE command, which ends the session.

Logging On

Logging on is the process of identifying and connecting the user to the computer. The :HELLO command, followed by the user and the account name, logs on the user who enters it. You are probably familiar with the :HELLO command in its simplest form, or perhaps in conjunction with one or more passwords. This section briefly describes the other options available by specifying optional parameters. Figure 8-1 shows the :HELLO command as it appears in the *MPE V Commands Reference Manual* (32033-90006).

```
:HELLO    [sessionname , ]username[ / userpass] : acctname[ / acctpass]
            [ , groupname[ / grouppass]]
  [ : TERM= { termtype }    ]
            { termname}
  [ : TIME=cpusecs ]
            {BS}
  [ ; PRI = {CS}   ]
            {DS}
            {ES}
  [ { ; INPRI = inputpriority}
    { ;  HIPRI  }
```

LG200077_054

Figure 8-1. The :HELLO Command

Required Parameters of the :HELLO Command

The :HELLO command has two required parameters: the user name and the account name. This means that you must specify a user and an account or the command will not work.

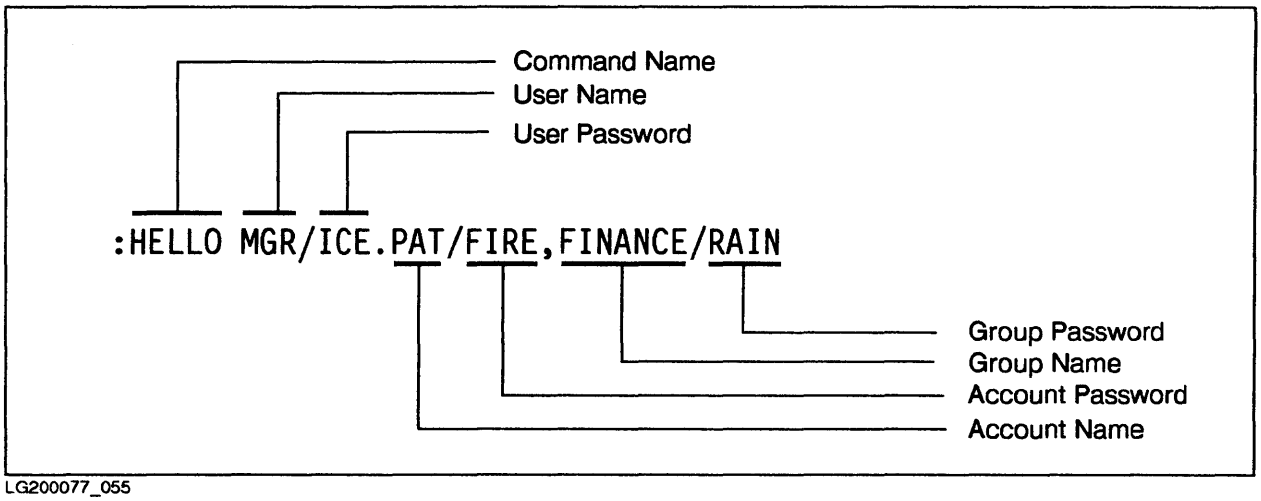
Optional Parameters of the :HELLO Command

Under most circumstances, only the user name and the account name are necessary to successfully log on. As with other commands, the system automatically assigns the most commonly used values, called "defaults," for all optional parameters.

Passwords

As shown in the syntax box in Figure 8-1, passwords can be assigned at the user, account, and group level. Figure 8-2 shows how the :HELLO command syntax would look with a password for the account, the user, or the group. However, since this logon sequence shows the passwords on the screen, someone standing nearby might see them. As an added security precaution, many systems ask (prompt) you for the password(s) after you enter the basic logon sequence. When you enter the password in response to a prompt, it is not displayed on the screen. This interaction is illustrated in Example 8-1.

Your manager should tell you what passwords apply to your account. For instructions on assigning and changing the user password, see Chapter 6, "Basic Account Structure".



LG200077_055

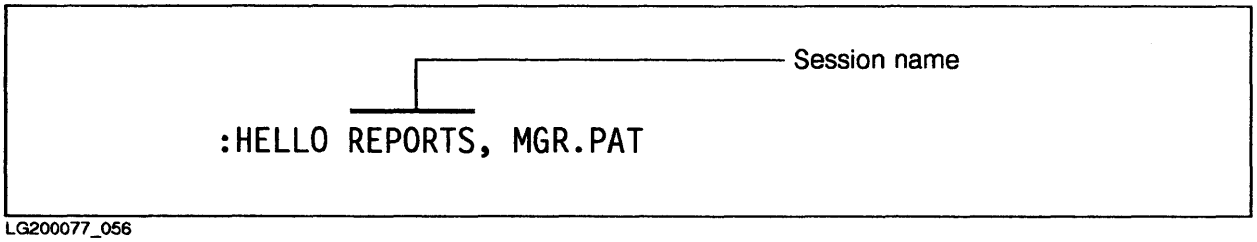
Figure 8-2. The Logon Passwords

NOTE

This illustration is for demonstration only. For security reasons, do not include passwords with the logon command by using the slash (/) character.

Session Name

You can also specify an optional session name before the user name, as shown in Figure 8-3. This is useful to identify a particular project or to identify an individual user when several people use the same account and user names. The :SHOWJOB command displays all session names, along with other information that shows what is happening on the system.



LG200077_056

Figure 8-3. The Optional Session Name

Other :HELLO Parameters

The parameters (TERM=, TIME=, PRI=, INPRI=, HIPRI=) are described briefly, although they are not used very often. Since these parameters are all keyword parameters, you can list them in

any order. For example, to specify the terminal and the input priority, either command would be valid:

```
:HELLO PAT.MGR;INPRI=13;TERM=18
```

or

```
:HELLO PAT.MGR;TERM=18;INPRI=13
```

The TERM= parameter is used mostly for terminals that are not Hewlett-Packard terminals. The most common is the designation TERM=18, which is a standard "dumb" ASCII terminal.

The TIME= parameter lets you limit the number of CPU seconds used by your session. After using the amount of time specified, MPE automatically logs off. This parameter is mostly used for jobs to prevent a program from accidentally tying up the CPU.

The PRI= parameter specifies your execution priority class, with BS being the highest possible value and ES the lowest. It should not be used by general users.

The INPRI= parameter allows you to set your "inpriority" setting, which determines whether or not you can get on the system. To successfully log on, this setting has to be higher than the jobfence setting assigned by the Operator. However, if the jobfence is set at the maximum priority, only System Supervisors and System Managers can log on, even if you specify the highest priority (INPRI=13).

NOTE

To find out what the settings for the outfence and the jobfence are, use the :SHOWJOB command. However, do not change your INPRI= setting without first checking with the operations staff.

The HIPRI= parameter lets users with special capabilities override the jobfence setting and the session limit.

The Basic Logon Procedure

To log on, you need to know the user name, account name, and any passwords that apply to the user or account. Your manager will give you this information. Also, if your company has more than one CPU connected by a PBX network, you will need to follow additional steps to connect to your particular system.

For a summary of the procedure that follows, refer to the brochure titled *HP 3000: Logging On and Off* (32650-90098). In addition to outlining the steps, it provides a place for you to write your logon sequence for quick, handy reference.

To log on:

1. Press [Return] for the colon (:) prompt.

   (If the prompt does not appear, press [Return] repeatedly. Make sure Remote mode is set to ON and Block mode is set to OFF, as described in "Checking Terminal Settings" in Chapter 2. If that does not work, ask for help.)

2. Enter HELLO, followed by the user and account names. If you make an error, use the [Back Space] key to erase to the error and then re-enter the command.

   ```
   :HELLO username.accountname
   ```

   The system displays either the logon message, indicating that you are logged on, or it prompts you for the password.

3. If the message ENTER ACCOUNT PASSWORD appears, enter the account password. For security reasons, passwords do not appear on the screen as you enter them.

4. If the message ENTER USER PASSWORD appears, enter the user password.

   The logon message appears.

The Logon Message

After you successfully log on, the logon message appears on the terminal screen. This message consists of two parts, the System Information and the Welcome message.

The System Information message includes the version of the MPE operating system, the date, and the time, similar to Example 8-1.

The Welcome message is programmed by the operations staff, so it varies from site to site and sometimes from day to day. This message often contains important system messages, such as scheduled down time or whom to call with problems.

Example 8-1 shows how the sample user MGR in the account PAT logs on, as well as a sample logon message. Since both names are associated with a password, MPE also prompts the user for passwords. These passwords are not displayed on the screen when they are entered.

```
:HELLO  MGR.PAT

ENTER ACCOUNT PASSWORD

ENTER USER PASSWORD


HP 3000/MPE V  G.A201 (BASE G.A2.01) THU, JUN16, 1987, 11:28 AM

    ********************************************************
                    WELCOME TO BITBANGER

      THE OPERATIONS STAFF WILL PERFORM A SYSTEM BACKUP AT
      6 P.M. TODAY. ALL USERS MUST BE LOGGED OFF BY THAT TIME.

      FOR QUESTIONS OR PROBLEMS, CALL X 767
    ********************************************************
```

Example 8-1. Sample Logon Procedure and Logon Message

Determining your Session Priority

The CPU has a limit as to how many jobs and sessions it can run at one time. Although the logon procedure usually works, at times the operations staff must limit access to the computer. Other system resources, especially printers, also have limits as to the amount of work they can handle. To protect the system from becoming overloaded, the system administrations staff has four tools: JLIMIT, SLIMIT, jobfence, and outfence.

The jobfence is the minimum priority required to log on for both jobs and sessions. The JLIMIT and SLIMIT determine the maximum number of jobs and sessions that can execute at the same time. Thus, if the SLIMIT is 80 and you are the 81st person trying to start a session, the terminal displays the following message:

```
CAN'T INITIATE NEW SESSIONS NOW.
```

If this happens, you either have to wait until another session logs off or until the Operator changes the SLIMIT setting.

In summary, you can log on if the number of sessions on the system does not exceed the SLIMIT, and if your input priority exceeds the jobfence. You can find out the settings for these values with the :SHOWJOB command (provided, of course, that you were able to log on).

Determining System Access (Input Priority)

The jobfence determines the minimum priority required to log on to the system, also known as the "input priority." Therefore, if the jobfence is set to 7, you need an input priority of at least 7 to log on. Since the usual default priority setting is 8, there should be no problem. However, during periods of unusually high demand, Operators may raise the jobfence. Note that the highest priority a user can request is 13. The priority 14 can be assigned only by the Operator, and the priority 15 is equivalent to the HIPRI priority setting.

The INPRI parameter in the :HELLO command lets you change your input priority to exceed the jobfence. However, since the operations staff usually has a good reason for raising the jobfence, do not change this setting without first checking with them.

To change your input priority:

Enter your logon sequence with the keyword parameter INPRI=, followed by the desired priority.

:HELLO user.account;INPRI=number (1-13)

Example 8-2 is identical to Example 8-1, except that the user changes the input priority to 13; the highest possible user-assignable value.

```
:HELLO  MGR.PAT;INPRI=13
ENTER ACCOUNT PASSWORD
ENTER USER PASSWORD

   HP 3000/MPE V  G.A201 (BASE G.A2.01) THU, JUN16, 1987, 11;28 AM

    ********************************************************
                    WELCOME TO BITBANGER

   THE OPERATIONS STAFF WILL PERFORM A SYSTEM BACKUP AT
      6 P.M. TODAY. ALL USERS MUST BE LOGGED OFF BY THAT TIME.

      FOR QUESTIONS OR PROBLEMS, CALL X 767
   ********************************************************
```

Example 8-2. Changing the Input Priority

Determining Output Priority

The outfence setting determines the required user priorities for using nonsharable output devices. There is no parameter in the :HELLO command for changing the user's outpriority.

However, you can control the priority of a printout by specifying the appropriate output priority in the DEV= parameter of a :FILE command. For a job as a whole, you can determine its output priority with the OUTCLASS parameter of the !JOB command. For details, refer to the *MPE V Commands Reference Manual* (32002-90003).

The :(COMMAND) LOGON Option

The :COMMAND LOGON option can cause a command or subsystem to be executed automatically when the user logs on. As soon as the command is executed or the user exits the subsystem, MPE automatically logs off. This procedure can save some time in logging on and entering a subsystem. It can also be used to limit some users' system access to a single subsystem.

To use the :(COMMAND) LOGON option:

In parentheses, enter the name of the command or subsystem you want to run, followed by the user and account names.

```
:(PROGRAM NAME) username.accountname
```

In response, the system displays the logon message, indicating that you are logged on. It then executes the command or runs the program named in the parentheses. If you specified a subsystem, you are ready to use that subsystem.

Example 8-3 shows how the user OLLIE in the account NOACCNT can go directly into the EDITOR subsystem. This example assumes that no passwords are associated with the account. Upon exiting the EDITOR subsystem, OLLIE is automatically logged off.

```
:(EDITOR.PUB.SYS)  OLLIE.NOACCNT

    ******************************************************
                welcome to the ACTION system

    ******************************************************

END OF PROGRAM

LOGON CHECK SUCCESSFUL

:HP32201A07.17 EDIT/3000 THU, JUN 16, 1987, 11;28 AM
(c) Hewlett-Packard Co. 1985
/
```

Example 8-3. Using the :(Command) Logon Option

Logging Off

Logging off with the :BYE command terminates the session and breaks the connection with the computer. The :BYE command has no parameters.

You should log off when you finish working to ensure that nobody accesses your account or files from your terminal. You should also log off whenever you leave your terminal for more than a few minutes. This secures your files and ensures your account is not charged for system usage.

To log off:

Enter BYE.

In response to the :BYE command, the system displays the information shown in Example 8-4. This includes CPU seconds used, number of minutes logged on, date logged off, and the time logged off.

```
:BYE

CPU=4      CONNECT=340          THU,JUN 20, 1987, 5:21PM
```

Example 8-4. Sample Logoff Procedure and Logoff Message

Correcting and Repeating Commands

The :REDO command lets you correct and repeat the last command entered without re-entering it.

If you make a mistake in entering a command, MPE displays an error message. With a short command, it may be easiest to enter the command again. However, some commands may be quite long, especially if you specify several parameters.

That is where the :REDO command saves time. First, it lets you correct errors in the last command you entered. Second, it repeats commands without reentering.

Unlike most commands, the :REDO command does not have parameters. Instead, it has the same subcommands as the /MODIFY command in the EDITOR subsystem. These commands let you insert, delete, and replace characters within the command line, or undo any editing changes.

Table 8-1 The :REDO Subcommands

Command	Function
R	Replaces the letter above it.
D	Deletes the letter above it.
I	Inserts the letter(s) that follows the I.
U	If used once, reverses last change. If used twice, reverts to the command as it was before it was changed with the :REDO command.

NOTE

You cannot use the :REDO command to correct the :HELLO command or any of its parameters. Unless you get it right, you cannot log on.

Correcting Commands with the :REDO Command

Using these subcommands is similar to using the EDITOR subsystem's /MODIFY commands.

To delete selected characters:

Use the space bar to move the cursor under character(s) you want to delete and enter D.

To insert characters:

Use the space bar to move the cursor under the character where you want to insert and enter I , followed by the character(s) you want to insert.

To replace characters:

Use the space bar to move cursor under the character(s) you want to replace and enter R, followed by the character(s) you want as replacements.

To cancel the most recent change:

Enter U once in the first character position in the line.

To cancel all changes made since the command was last executed:

Enter U again in the first character position.

Example 8-5 shows how to replace the "V" character in the file name with an "F," and how to delete the extra "C" character in "ASCII." For the sake of illustration, it then shows how to undo the last change made. Note that you must make corrections one at a time.

```
:REDO

:BUILD NEWVILE;REC=80,1,F,ASCCII
        RF

:BUILD NEWFILE;REC=80,1,F,ASCCII
                           D

:BUILD NEWFILE;REC=80,1,F,ASCII
 U

:BUILD NEWFILE;REC=80,1,F,ASCCII
```

Example 8-5. Using the :REDO Command to Correct Errors

Repeating Commands with the :REDO Command

You can also use the :REDO command to repeat the previous command. This can save time and CI errors, especially with commands that have many parameters.

To repeat commands:

1. Enter the command.
2. Enter REDO and press Return twice.

The first time you press Return, MPE displays the command on the screen. The second time, it executes the command.

Running Programs and Subsystems

A major reason for initiating sessions is to run programs. These programs and subsystems perform the tasks necessary for the work you do. The :RUN command, followed by the name of the program or subsystem, actually runs the program. If the program is in a group or account other than your home group, you must also enter the group and account names in which the program resides.

Advanced users can also specify a number of optional parameters that expand the control over how the program executes, including a program entry point, the maximum stack area permitted, and many more. For complete information, refer to the *MPE V Commands Reference Manual* (32022-90006).

Example 8-6 shows how to run the SPOOK subsystem to access spoolfiles. If the system manager created a UDC for this command, users could run the subsystem by entering a simpler command.

```
:RUN  SPOOK5.PUB.SYS

:SPOOK5  G2.02.B0              (c) HEWLETT-PACKARD CO. 1983
>
```

Example 8-6. Running Programs and Subsystems

Interrupting Commands and Programs

In some subsystems, you can enter MPE commands directly without leaving the subsystem by entering a colon (:) after the subsystem prompt. For example, to execute a :LISTF command in EDITOR, you would enter :LISTF at the slash (/) prompt.

Most subsystems and programs do not execute MPE commands from within. However, you can use the Break/Reset key to interrupt most commands, programs, and subsystems. In this way, you can interrupt one program or command, enter MPE commands, and then resume at the point you left off.

NOTE

Not all commands can be interrupted with Break/Reset, and some commands cannot be executed while another program is suspended. Appendix A lists those commands that can be interrupted (breakable commands), as well as those commands that work while another command or program is suspended (executes in break).

Assume, for example, that you are running the LISTDIR5 program. You want use the >LISTSEC command to find out the security levels for a given file. However, you do not remember the file's name, which is a required parameter of the >LISTSEC command.

If you try to enter :LISTF at the chevron (>) prompt, MPE displays the following message:

```
**MPE COMMANDS MAY NOT BE EXECUTED FROM LISTDIR5
```

However, by interrupting the LISTDIR5 subsystem, you can execute a :LISTF command to find out the file name. You can then return to the LISTDIR5 subsystem and enter >LISTSEC and the file name to display the security levels set for that file.

This section explains how to interrupt subsystems without leaving the subsystem, how to interrupt a program and then resume it, and how to stop a program's execution altogether.

CAUTION

Do not use Break/Reset routinely to exit from programs.

Using Break/Reset with applications like HPSLATE or HPDESK can destroy data. Also, do not use this key with any "full screen" applications that use Block Mode, such as VPLUS.

Unlocking the Keyboard

In addition to interrupting commands and programs, Break/Reset can also reset a keyboard that does not respond. After checking that the keyboard cable is firmly plugged into the terminal, you can execute either a "soft reset" or a "hard reset." A soft reset will not destroy information on the screen. A hard reset, on the other hand, has the same effect as turning the power off and then back on. Any information on your screen is lost, but you are still logged on.

To unlock the keyboard with a soft reset:

Press CTRL and Break/Reset simultaneously.

To unlock the keyboard with a hard reset:

1. Press SHIFT, CTRL, and Break/Reset simultaneously.
2. If this does not work, contact your System Manager.

Interrupting a Subsystem

You can interrupt a subsystem to return to the subsystem prompt with the CTRL Y command sequence. For example, if you entered the /LIST ALL command in the EDITOR subsystem, you can stop the resulting listing.

To interrupt a subsystem:

Press CTRL Y.

The subsystem is interrupted and the subsystem prompt appears.

Temporarily Interrupting a Program

You can temporarily interrupt a program to use MPE commands. When you are finished with MPE, you can return to the program with the :RESUME command.

To temporarily interrupt a program:

1. Press Break/Reset.

   The system displays the colon (:) prompt.

2. Enter the desired MPE commands.

   The system executes the MPE commands.

3. Enter RESUME to return to the program at the point you left it.

   The system displays READ pending. The subsystem prompt (if any) reappears after you press Return.

Aborting a Program

You can stop a program's execution altogether, which is known as "aborting" the program.

To abort a program:

1. Press Break/Reset.

   The system displays the colon (:) prompt.

2. Enter ABORT to stop the program's execution.

   ```
   :ABORT
   ```

   The system displays PROGRAM ABORTED PER USER REQUEST and redisplays the colon prompt.

Monitoring the Session

When you work on the terminal, you usually get results right away, even if it is only an error message. But when you use some of the system's other resources, you cannot see as easily what is going on.

Assume that you want to print a file. Your file may print right away or it may have to wait its turn on the printer. Luckily, you do not have to camp out at the printer to wait for your printout, because MPE's commands let you monitor its progress.

The monitoring commands give you access to a great deal of information about what is going on in the system, ranging from the time of day to the device numbers assigned to your terminal and those of other users.

This section describes the following commands: `:SHOWTIME`, `:SHOWALLOW`, `:SHOWME`, `:SHOWDEV`, `:SHOWJOB`, and `:SHOWOUT`.

Displaying the Current Date and Time

In response to the `:SHOWTIME` command, the terminal displays the current time as it was set by the Operator. Besides checking to see whether it is time for lunch, this command is useful to clock the time it takes to run a program. The `:SHOWTIME` command has no parameters.

To display the current date and time:

Enter `SHOWTIME`

In response, a display similar to Example 8-7 appears.

```
:SHOWTIME

TUE, JUN 14, 1987,   1:33 PM
```

Example 8-7. Sample :SHOWTIME Display

Displaying Availability of Special Commands

Some commands can only be executed by certain users. To find out what special commands you or other users in your group may be allowed, enter `:SHOWALLOW`. This command lists the session numbers of active sessions and their user and account names. The `:SHOWALLOW` command has no parameters.

To find out what special commands are available to you:

Enter SHOWALLOW

In response, a display similar to Example 8-8 appears.

```
:SHOWALLOW

#S513      MGR.PAT

USER HAS THE FOLLOWING COMMANDS ALLOWED:

ABORTIO        DSCONTROL         ABORTJOB             ALTSPOOLFILE
LIMIT


THE FOLLOWING HAVE BEEN GLOBALLY ALLOWED:

ABORTIO        DSCONTROL         ABORTJOB             ALTSPOOLFILE
LIMIT
```

Example 8-8. Sample :SHOWALLOW Display

Displaying Information about the Current Session

The :SHOWME command has no parameters. It displays information about the current session, including:

- The user name and account name (line 1)
- The name of the group you are accessing (line 1)
- The date and time (lines 3 and 4)
- The device number of the terminal (line 6)
- The Welcome message, if any, that appeared when you logged on (line 8)

To display information about the current session:

Enter SHOWME

In response, a display similar to Example 8-9 appears.

```
:SHOWME

USER: #S513, MGR.PAT,PUB              (IN BREAK)
MPE VERSION:  HP 32033G.A2.01    (BASE G.A2.01)
CURRENT:  TUE,JUN 14,1987,  1:52 PM
LOGON:    TUE,JUN 14,1987, 11;59 AM
CPU SECONDS:12                              CONNECT MINUTES 154
$STDIN LDEV:55                              $STDIN LDEV: 55


     ********************************************************
                      WELCOME TO BITBANGER

      THE OPERATIONS STAFF WILL PERFORM A SYSTEM BACKUP AT
      6 P.M. TODAY. ALL USERS MUST BE LOGGED OFF BY THAT TIME.

      FOR QUESTIONS OR PROBLEMS, CALL X 767
     ********************************************************
```

Example 8-9. Sample :SHOWME Display

Displaying Information about System Devices

When entered without parameters, the :SHOWDEV command displays the status of all devices attached to the HP 3000. These devices are identified by their device identification numbers (ldev), as shown in Example 8-10.

To display the status of all devices:

Enter SHOWDEV

In response, a display similar to Example 8-10 appears.

```
:SHOWDEV

LDEV       AVAIL      OWNERSHIP   VOLID DEN  ASSOC.

  1        DISC (RPS) 26 FILES
  2        DISC (RPS) 32 FILES
  3        DISC (RPS) 53 FILES
  4        SPOOLED
  5        AVAIL
  6        SPOOLED    SPOOLER OUT
  7        A UNAVAIL  #S202: 2 FILES
  8        A UNAVAIL  #S423; 8 FILES
  9        AVAIL
```

Example 8-10. Sample Partial :SHOWDEV Display

To display the status of a particular device:

Enter SHOWDEV, followed by the device's logical device (LDEV) number.

Example 8-11 shows how to display the status of a terminal that has been assigned the ldev number 123.

```
:SHOWDEV 123

LDEV       AVAIL              OWNERSHIP   VOLID DEN  ASSOC.

123        A UNAVAIL          #S1051: 2 FILES
```

Example 8-11. Sample :SHOWDEV Display, with Parameter

Displaying Information about Jobs and Sessions

The :SHOWJOB command, which displays information about jobs and session executing on the system, is one of the most commonly used commands.

Depending on which parameters you specify, you can display information about the following:

- All jobs and sessions (:SHOWJOB)
- A particular job or session (:SHOWJOB JOBNUM)
- The number of jobs and sessions in each processing state, as well as the current jobfence and job and session limits. (:SHOWJOB STATUS)
- Scheduled jobs only (:SHOWJOB SCHED)

To display information about scheduled jobs only:

Enter SHOWJOB SCHED

In response, the system displays a screen similar to Example 8-12.

```
:SHOWJOB SCHED

CURRENT: 7/26/87 13:02

JOBNUM     STATE   IPRI    JIN     JLIST   SCHEDULED-INTRO   JOB NAME

#435       SCHED   15      10S     LP      7/26/87   14:21   JOBCHECK,RSPOOL.SYS
#394       SCHED    8      10S     LP      7/26/87   20:00   SWING4,Operator.SYS

2 SCHEDULED JOB(S)
```

Example 8-12. Sample :SHOWJOB SCHED Display

To display information about all jobs and sessions:

Enter SHOWJOB

In response, the system displays a screen similar to Example 8-13. Each column shows a particular piece of information. As you read the descriptions of each column, refer to this display.

```
:SHOWJOB

JOBNUM        STATE  IPRI    JIN   JLIST  INTRODUCED  JOB NAME

#S202         EXEC          20    20     MON 11;38A  CONSOLE.SYS
#S367         EXEC          142   142    TUE  6:14A  JAN,ART.SYS
#S540         EXEC   QUIET  35    35     TUE  9:20A  MARY.JONES
#J216         EXEC          10S   LP     TUE 10:30A  MAIL,Y,OFFICE
#J217         WAIT          10S   LP     TUE 12:25A  STATS.ACCT
#S582         INTRO         55    55     TUE  1:02A  PAT.MGR

6 JOBS:
   1 INTRO
   1 WAIT
   4 EXEC; INC 3 SESSIONS
   0 SUSP
jobfence=6; JLIMIT=11; SLIMIT=50
```

Example 8-13. Sample :SHOWJOB Display, without Parameters

JOBNUM Column

Lists the number assigned by the system when you start a session. Sessions are preceded by the letter S, jobs by the letter "J". The numbers that follow are assigned in the order that people log on or submit jobs and continues until there is a COOLSTART. At that point, session and job numbers start again from 1.

STATE Column

Displays what state a session or job is in. When you first enter the :HELLO command, before MPE has completed checking the logon sequence against its directory, INTRO appears briefly in this column. Whenever you successfully log on, the session is in the EXEC (executing) state.

Note that jobs may also have WAIT, SUSP, or SCHED states (see Chapter 9, "JOBS").

IPRI Column

If this column is blank for any given session, you can send messages to that session with the :TELL command. If QUIET appears in this column, only WARN messages sent by the Operator will get through.

JIN and JLIST Columns

Show the ldev numbers assigned to the input (JIN) and output (JLIST) devices. Note that for sessions, those numbers are the same--the device number assigned to the terminal.

INTRODUCED Column

Shows the date and time the user started the session (or job).

JOBNAME Column

Shows a session's user and account name, as well as the session name.

Summary Report

In addition to the information in these columns, the :SHOWJOB command displays information about the number of jobs and sessions that are running, as well as the settings for the jobfence and the SLIMIT and JLIMIT (described in "Determining Your Session Priority"). In addition, it shows any scheduled jobs.

If your session's priority does not exceed the jobfence when you try to log on, or if it is over the SLIMIT setting, the system will display

```
CAN'T INITIATE NEW SESSIONS NOW.
```

Displaying Information about Spoolfiles

To understand the :SHOWOUT command, it might be helpful to review a processes called "spooling."

Discs are called "sharable devices" because more than one user can access them at the same time. Devices such as printers and tape drives can handle only one process at a time; therefore, they are called "non-sharable devices." In a way, they are similar to single-user telephone lines. Only one person at a time can use the line; all others get a busy signal.

The spooling process manages the input and output on nonsharable devices by assigning requests in the order they are received. These requests then wait their turn to be processed.

The spooling process handles both information going to the computer from nonsharable devices (usually terminals) as well as information from the computer to nonsharable devices (usually printers).

The :SHOWOUT command displays information about spoolfiles. This allows you to see what printouts you requested and where they are in the printing process. You can also display information for spoolfiles not accessible to you, including:

- The state of all spoolfiles on the system (:SHOWOUT STATUS)
- Spoolfiles in the current session (:SHOWOUT)

To see a summary of spoolfiles:

Enter SHOWOUT STATUS

In response, the system displays a screen similar to Example 8-14. It shows how many spoolfiles there are, their state, and the total number of sectors they take up.

```
:SHOWOUT STATUS

40 FILES:
     0 ACTIVE
     3 READY; INCLUDING 3 SPOOLFILES, 3 DEFERRED
     37 OPENED; INCLUDING 4 SPOOLFILES
     0 LOCKED; INCLUDING 0 SPOOLFILES
     7 SPOOLFILES:  44250 SECTORS
outfence = 8
```

Example 8-14. Sample :SHOWOUT STATUS Display

To see what spoolfiles exist for your session:

Enter SHOWOUT

In response, the system displays a screen similar to Example 8-15. Like the :SHOWJOB command, each column shows a particular piece of information. As you read the descriptions of each column, refer to this display.

```
:SHOWOUT

DEV/CL     DFID       JOBNUM     FNAME        STATE      FRM SPACE RANK PRI #C

33         #0118      #S513      $STDLIST     OPENED


outfence = 6
```

Example 8-15. Sample :SHOWOUT Display

DEV/CL Column

Lists the logical device number (DEV) or the device class name (CL) of the output device on which a file will be printed. These values are assigned by the System Manager.

The DFID (Device File Identification Number) Column

The spooler program assigns a unique DFID number to each output request, in the order received. This number is preceded by an O (for output).

JOBNUM Column

Lists the number of the session or job that requested the printout. This helps the Operator to identify which printouts belong to which user.

FNAME Column

Lists the file name assigned to the temporary spooling file created for each output file. The default name is `$STDLIST`, which means that the file will be printed on the default output device.

STATE Column

Shows where the listing is in the printing process. There are four possibilities, `ACTIVE`, `OPENED`, `READY` or `LOCKED`.

`ACTIVE` means that the file is being printed when the `:SHOWOUT` command was entered.

`OPENED` means the spooler process is still assembling that file. It is not ready to be printed.

`READY` means the file is completely spooled and ready to be printed. Files in the `READY` state can be accessed with the SPOOK subsystem.

`LOCKED` means that the file is ready to be printed, but that someone is accessing that file with the SPOOK subsystem.

FRM Column

Used for special forms printing. If the letter "F" appears, it means that a forms alignment message applies to this device.

SPACE Column

Lists the approximate disc space currently used (in sectors).

RANK Column

Lists the output file's rank on the system, which is determined by the spooler process. The time, priority setting, and the logical device number (or class name) help determine the order in which files are printed. If a "D" (for deferred) appears in this column, it means that file cannot print at this time because its priority is less than or equal to the outfence setting.

PRI Column

Shows the file's requested output priority. (The default is 8. The lowest possible priority is 1; the highest 13.)

#C Column

Shows the number of copies requested.

Summary Report

Shows the outfence setting assigned by the Operators. The outfence determines who can get printouts. It is similar to jobfence in that it is a number set by the Operator to control the number of people using the printer or other nonsharable output devices.

If your output priority is too low, your printout will not print. You can either wait until the Operator lowers the outfence, change your priority with the >ALTER command in the SPOOK subsystem, or redefine the OUTCLASS parameter in the :JOB command.

Sending and Receiving Messages

As a general user, you will be on the receiving end of most messages. This may include messages from the Operator or from other users. This section explains the most common kinds of messages you might expect to receive. It also tells you how to keep other users from interrupting you with messages. In addition, it explains how to send messages to the Operator or to other users.

Messages from the Operator

The two most common types of messages from the Operator include the Welcome message and the WARNING message.

The Welcome Message

The Welcome message appears on the screen whenever you log on. Since this message was generated by the Operator, there is nothing you can do to change it. However, do not ignore it. This is often the best way to get important system messages, such as scheduled down time or database problems.

The WARNING Message

The WARNING message will interrupt any job or session, even if you used the :SETMSG command to keep other users from interrupting your work. Operators usually use this command in an emergency or just before the system goes down.

Example 8-16 shows a typical WARNING message.

```
OPERATOR WARNING: Please log off.
```

Example 8-16. Sample WARNING Message from Operator

Sending Messages

You can send messages to the Operator or to other users. You can also use the :SETMSG command to prevent messages from other users from being displayed on your terminal.

The :TELLOP Command

The :TELLOP command gives you a chance to talk back to the Operator by sending a message to the System Console. To alert the Operator, you can include a CTRL G code in the message to cause the Console to beep. You may also want to follow it up by a call to the computer room.

To send a message to the Operator:

Enter TELLOP, followed by the message.

The message in Example 8-17 asks the Operator to get a tape ready. Pressing CTRL G causes the Console to beep when the message is received.

```
:TELLOP   PLEASE MOUNT TAPE #NL1652 ON TAPE DRIVE
```

Example 8-17. Sample Message to Operator

The :TELL Command

You can send messages to other users by entering the :TELL command, followed by their session number (which you can find out by using the :SHOWJOB commands). To increase your chances of getting their attention, you can incorporate the CTRL G code, which will cause the terminal to beep.

To send a message to other users:

Enter TELL, followed by the session number and the message.

Example 8-18 shows how to use the :TELL command to send a message to another user.

```
:TELL #S54   ONLY TWO MORE REVIEW DAYS TILL DEADLINE
```

Example 8-18. Sample Message to Other User

If your message calls for a response and you get none, use the :SHOWJOB command, find the session number of your target audience, and check to see if the word QUIET appears in IPRI column. If it does, it means that person is blocking messages, so you will have to try a telephone call or a tap on the shoulder.

Turning Messages On and Off

The :SETMSG command keeps other users from interrupting you with messages initiated by the :TELL command. No messages, except WARNING messages from the Operator, can get through. When you turn off messages, you will be in "quiet mode," as indicated by the word QUIET in the IPRI column of the :SHOWJOB display.

NOTE

Some applications, like HPDRAW and HPSLATE, automatically put you in quiet mode.

To block messages:

Enter SETMSG OFF

The word QUIET appears in the IPRI column in the :SHOWJOB display (see Example 8-13).

To receive messages again:

Enter SETMSG ON

Other users can again send you messages with the :TELL command. The word QUIET disappears from the IPRI column in the :SHOWJOB display (Example 8-13).

Using the HELP Facility

The online HELP facility displays all commands, associated parameters, and usage examples. For a detailed description and procedures, refer to Chapter 3, "Introduction to Commands."

Session Security

Chapter 6, "Account Structure" describes how to use lockwords and file access provisions to protect your files. In addition, you should protect your files whenever you physically leave the terminal by logging off. Otherwise, anyone can read, change, or erase your files.

9

Jobs

Overview

Batch processing allows you to submit a sequence of commands and information to the computer as a single package. This package is called a "job."

This chapter contains the following information about jobs:

- Introduction
- The batch processing commands
- Creating a job file
- Submitting the job file
- Monitoring a job
- Security of batch data
- Information on the job printout

Introduction

Batch processing refers to a processing method where all commands and data are submitted to the computer in one piece, called "job file" or "job", for short. Sometimes jobs are also referred to as "JCL" (job control language)," or "job stream,". Submitting a job to be processed is referred to as "streaming" a job. For an illustration that compares the concept of jobs and sessions, see Example 1-7.

For many tasks, batch processing is much more efficient than the interactive processing of sessions because it does not require continuous input or supervision. Once you create a job file with a program such as EDITOR, that file is stored on disc. You can then process all the commands in that job file at any time with a single command. Your terminal is free for other work while the job is processed. Any output from the job is printed on the printer instead of being displayed on the terminal. This is an advantage if you need printouts to refer back to later.

You should also use a job if your task requires a lot of system resources. Since jobs generally have a lower execution priority than sessions, this prevents your work from tying up system resources. Large jobs are often scheduled to run overnight or on weekends. Such jobs are also called "background jobs." Check with your System Manager for guidelines for your system. By scheduling jobs, you can control when your jobs are run.

Before you can run a job, you must first log on. To do so, follow the procedures in Chapter 8, "Managing Sessions."

For additional information about jobs, refer to the *System Operation and Resource Management Reference Guide* (32033-90005) and the *Guide For the New Operator* (32033-90021)

The Batch Processing Commands

You can include any MPE commands in job files to be executed. In addition, there are commands that are used only in jobs.

Job files must begin with the `!JOB` command and end with the `!EOJ` command. In addition, the optional commands `!COMMENT` and `!CONTINUE` commands are frequently used in jobs, as well as the message commands `:TELL` and `:TELLOP`. The `:STREAM` command submits the job to the system to be processed. These commands are summarized in Table 9-1.

Table 9-1. Batch Processing Commands

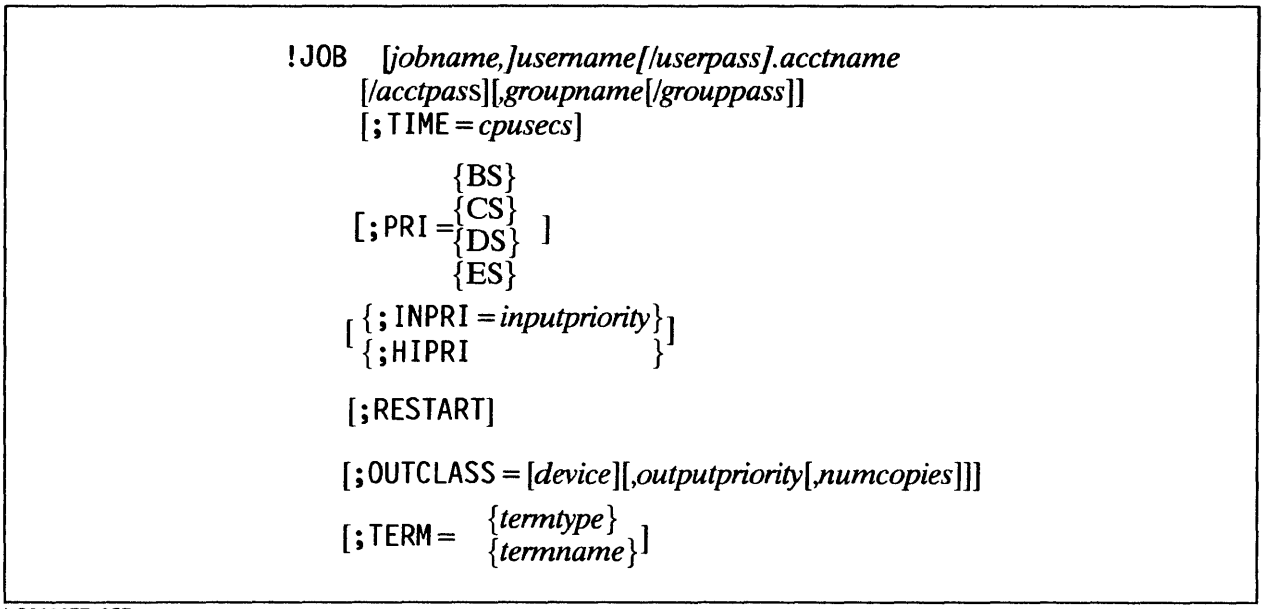
Command	Function
!JOB	First command in a job file. Requires the user and account name. It can also include other optional parameters.
!COMMENT	Lets you include explanatory comments, notes, or headings.
!TELL	Displays a message at a user terminal.
!TELLOP	Displays a message at the Operator's Console.
!CONTINUE	Prevents the command that follows it from aborting the job.
!EOJ	Terminates the job.
:STREAM	Submits the job file for processing.

The Prompt

In jobs, you must enter a character that simulates the colon (:) prompt before each command to be executed. The exclamation mark (!) is the most commonly used symbol. Unless you specify otherwise, MPE expects this symbol in column 1 of each line in the job file, before any MPE commands.

The !Job Command

The !JOB command signals MPE that the commands that follow should be executed as a batch. The !JOB command must also include information that identifies the person who created the job file as a legitimate user. For that reason, its syntax, which is shown in Figure 9-1, is very similar to that of the :HELLO command.



LG200077_057

Figure 9-1. The !JOB Command

Like the :HELLO command, the !JOB command has two required parameters: the user name and the account name. Therefore, you must specify a user and account name with the !JOB command.

Optional Parameters of the !JOB Command

Under most circumstances, only the user and account name are necessary to successfully stream a job. As with other commands, the system automatically assigns default values for all optional parameters.

The optional !JOB parameters include the positional parameters *jobname, userpass, acctpass,* and *grouppass.* In addition, the !JOB command contains the keyword parameters TIME=, PRI=, INPRI=, HIPRI=, RESTART, and OUTCLASS=. As you know, you must enter positional parameters in the order shown. Keyword parameters, on the other hand, can be listed in any order. For example, to specify the terminal, the inputpriority, and the RESTART parameter, either command would be valid:

```
!JOB MGR.PAT;INPRI=13;OUTCLASS=PP;RESTART
```

or

```
!JOB MGR.PAT;OUTCLASS=PP;RESTART;INPRI=13
```

The Positional Parameters

The optional parameters that specify passwords are not commonly used in logon sequences for security reasons. Unlike sessions, where MPE prompts for passwords, job files must contain any

associated passwords or the job will not work. Figure 9-2 shows a sequence that contains all possible passwords. (For instructions on assigning and changing user passwords, see Chapter 6, "Basic Accounting Structure".)

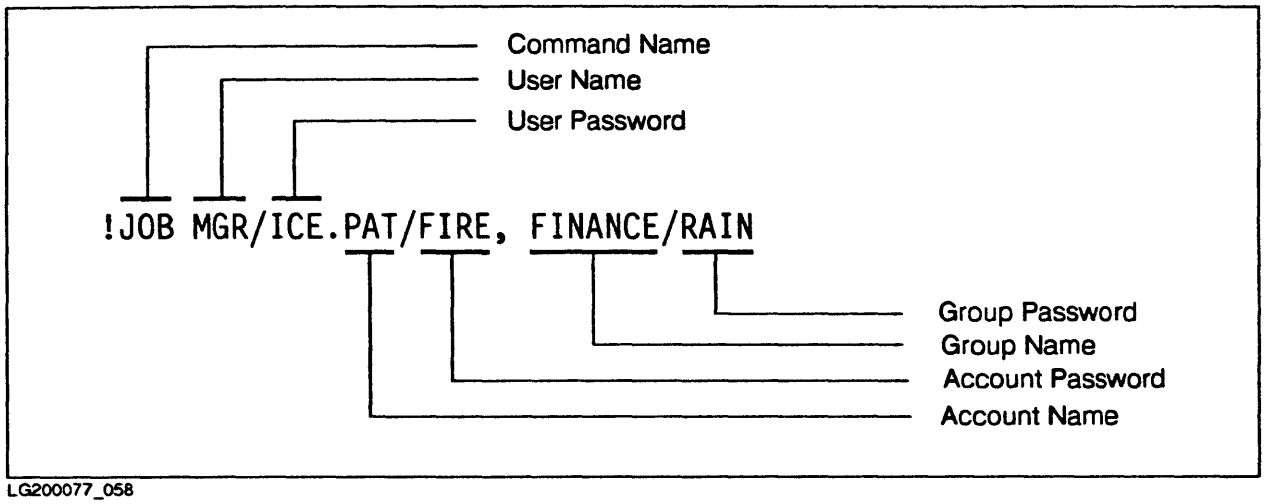


Figure 9-2. The Passwords

For security reasons, however, many sites have created special programs to insert passwords into job streams, so that passwords do not have to be stored with the job file. Check with your manager for the correct procedure for using passwords in jobs.

For an additional security measure, see the section "Security of Batch Data."

NOTE

The HP Security Monitor (HP30392A) is a product that may be purchased separately. It solves the security problem posed by including passwords in a job stream.

Job Name

To identify a particular job, you can also specify an optional job name before the user name, as shown in Figure 9-3. This may be helpful if you run several jobs.

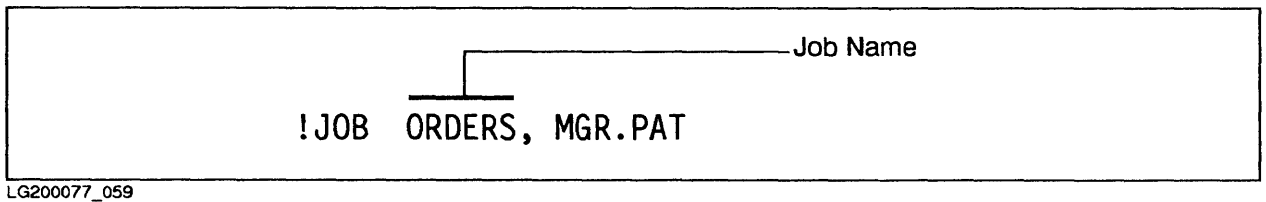


Figure 9-3. The Job Name

NOTE

Don't confuse the optional *jobname* parameter with the job's file name. The file name is required when you stream the job. (Of course, you could use the file name as the optional job name, which would eliminate any possible confusion.)

The Keyword Parameters

This section briefly describes the remaining parameters in the !JOB command.

The TIME= parameter lets you limit the number of CPU seconds used by your job. After using the amount of time specified, MPE automatically stops the job, whether or not it has finished processing the job. For example, someone testing a new program might add the TIME= parameter, along with a reasonable time limit, in case the program gets caught in a loop.

The PRI= parameter lets you set your execution priority. The default for jobs is DS. You can change this priority by specifying this parameter in the !JOB command, in a way similar to Example 8-2 in Chapter 8, "Managing Sessions." However, do not change the execution priority of your job without first checking with the operations staff.

The INPRI= parameter allows you to set your "inpriority" setting, which must exceed the jobfence set by the operations staff to limit the number of jobs and sessions on the system. Depending on the company, this setting may be used to specify "overnight" or "weekend" jobs. In any case, your job will be copied to the spooling disc, where it waits its turn in the WAIT state. If the jobfence is set at the maximum 13, only System Supervisors and System Managers can start jobs, even if you specify INPRI=13.

The HIPRI parameter lets users with special capabilities override the jobfence and JLIMIT setting.

NOTE

To find out what the settings for the jobfence are, use the :SHOWJOB command.

However, since the operations staff usually has good reasons for raising the jobfence to keep users off the system, don't change your PRI or INPRI settings without first checking with them.

If the system crashes, the RESTART parameter automatically restarts a job that was executing when the Operator performs a WARMSTART to bring the system back up. The job is restarted from the beginning, but no work already done is undone. As a result, jobs must be carefully designed to be restartable in this way.

The OUTCLASS parameter lets you specify the device where you want the job's output printed, the job's output priority, and the number of copies to be printed.

Example 9-1 shows a !JOB command that includes the user name MAC and the account name TECHPUB. In addition, it specifies a user password SUPER, the RESTART parameter, and the OUTCLASS parameter to request five copies on device LP. Note that the OUTCLASS parameter also incudes a place-holding comma for the omitted subparameter *outputpriority*.

```
!JOB MAC/SUPER.TECHPUB; RESTART; OUTCLASS=LP,,5
```

Example 9-1. Sample !JOB Command, with Parameters

The !COMMENT Command

The !COMMENT command lets you include comments or notes within the file. These comments can create headings or explain the purpose of commands or logic used. This is particularly helpful if other users use your job file for their own purposes.

The !TELL Command

The !TELL command sends a message to a specified user terminal. You can use this command to send a message to another user when your job reaches a certain point in its execution, or to send a progress report to your own terminal. For example, you may send a message to your own terminal if you want to know when the job finishes executing.

The !TELLOP Command

The !TELLOP command sends a message to the System Console. This message is preceded by the time it was transmitted by your job/session number. You can use this command to send a message to the Operator at a specified time while your job is being processed.

The !CONTINUE Command

In a session, a command that contains an error results in an error message from the CI (Command Interpreter), which is displayed on the screen. In a job, such an error causes a job to abort. An error message is displayed on the screen and printed on the printout.

The !CONTINUE command allows a job to continue to be processed if it appears immediately before a command that contains an error. Therefore, you can use the !CONTINUE command before any command you suspect of causing a problem. The job will continue to run even if an error occurs, and the job printout will contain the error message where the error occurred.

NOTE

Use the !CONTINUE command carefully. There are times when it would be better to let a job abort rather than ignore an error.

The !EOJ Command

The !EOJ command terminates a job and displays the following information on the job's printout: CPU time (in seconds), time elapsed since the beginning of the job (rounded to the nearest minute), and the date and time.

Working with Jobs

Working with jobs consists of two basic steps:

- Creating a job file that contains the necessary commands and data.
- Submitting that job file for processing with the :STREAM command.

Creating a Job File

This section describes how to create a job file, including the commands used to initiate and control jobs. The !JOB and !EOJ commands are required to begin and end each job. The !COMMENT, !TELLOP, and !CONTINUE commands are optional.

Once you create a job file with these commands, you can run it as often as you want with a single command (:STREAM jobfile). Any output produced is automatically printed on the printer.

The first step is to assemble all the commands and data into a job file, using EDITOR or your favorite word processing program. A job file consists of three basic parts:

1. The !JOB command and associated parameters to log on the system and initiate the job.
2. An assortment of MPE or subsystem commands and data.
3. The !EOJ command to signal the end of the job file.

Figure 9-4 shows a simple job file named DAILY. You could print the contents of this file at any time with the command :STREAM DAILY.

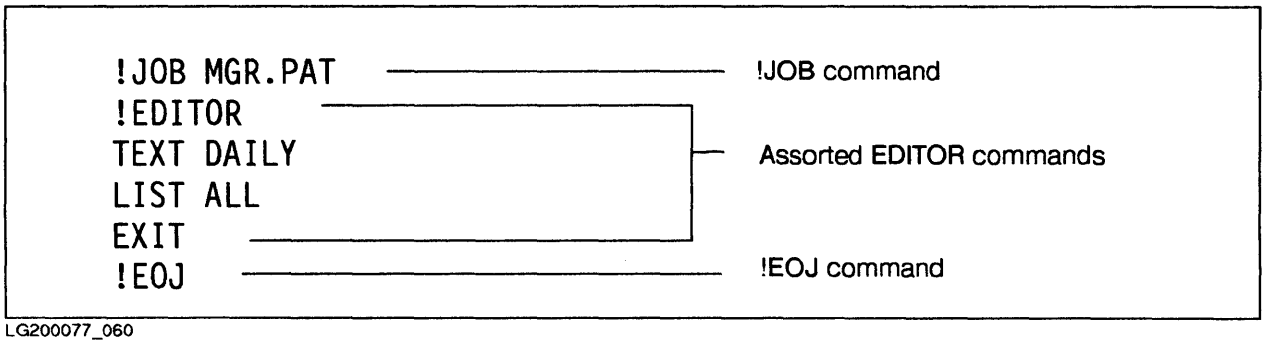


Figure 9-4. Anatomy of a Job File

You can create a job file with any word processing program that creates ASCII files. Since this manual describes the EDITOR subsystem, this section shows how to create a job file with EDITOR. Once created and saved with the /KEEP command, the job file is stored on disc. If needed, you can then easily modify the file. Note also that the system substitutes a colon (:) for the exclamation point (!) in the job listing.

NOTE

Exception: Files created with HPSLATE must be converted to EDITOR files before they can be used as job files. This conversion is one of the options offered in HPSLATE's menu.

MPE accepts almost any MPE commands or subsystem commands in jobs. For example, you can use the :RUN command to run subsystems like LISTDIR5 or SORT-MERGE, you can write file equations, or you can execute monitoring commands like :SHOWJOB or :SHOWME.

To create a job file:

1. Enter `EDITOR` to start the EDITOR subsystem.
2. Enter `ADD` or `A` to open a new document.
3. Enter `!JOB,` followed by any passwords and optional parameters.

   A new numbered line appears after you press Return.
4. Enter the next command that you want to include in the job file. Be sure to precede all MPE commands with the exclamation (!) prompt.

NOTE

If the command is a subcommand of one of MPE's subsystems, do not use the exclamation (!) prompt before the subcommand.

---

5. Repeat Step 4 for all commands to be executed in the job.
6. Enter `!EOJ` to signal the end of the job file.
7. Enter two slashes (//).

   The system displays three dots and the slash prompt.
8. Enter `/KEEP,` followed by the file name of the job file.

Example 9-2 shows the creation of a sample job file called JSFFINAL. This job includes the RESTART parameter, which means the job will start again if the Operator does a WARMSTART. It also includes the OUTCLASS parameter that defines which device the output will be printed on, and the number of copies to be printed.

The job itself runs the SORT program to sort, alphabetically by last name, the file created in Example 5-9 in Chapter 5. (To see how the same task is accomplished in an interactive session, see Example 5-10 in Chapter 5.) When the job is completed, the terminal displays the message JSFFINAL FINISHED.

```
:EDITOR
/

/ADD

1       !JOB JSFINAL,MGR.PAT; RESTART; OUTCLASS=PP,,15
2       !COMMENT    JOB SORTS SAN FRANCISCO EMPLOYEES BY LAST NAME
3       !RUN SORT.PUB.SYS
4       INPUT SFLIST
5       OUPUT SFFINAL
6       KEY 1, 10
7       END
8       !EDITOR
9       T SFFINAL
10      L ALL
11      EXIT
12      !TELL MGR.PAT  JSFFINAL FINISHED
13      !EOJ
14      //

...

/KEEP JSFFINAL

/EXIT

END OF PROGRAM
:
```

Example 9-2. Creating a Job File with EDITOR

Submitting (Streaming) the Job File

Once you have created a job file, you can submit it to the system for processing with the :STREAM command. If you enter the :STREAM command without the optional parameters, the job is spooled to a disc file, assigned a job number, and put in line (according to its priority) to be processed and printed. If you specify the optional parameters, you can schedule the job to run at a specific time or date. For further information, consult the *MPE V Commands Reference Manual* (32033-90006).

To submit a job file for processing:

Enter :STREAM, followed by the filename.

```
:STREAM filename
```

The terminal displays the job number assigned to your job by MPE.

If the message STREAM FACILITY NOT ENABLED: SEE OPERATOR appears, ask your Operator to start the streaming facility on the system console.

Example 9-3 shows how to stream the job file SFFINAL.

```
:STREAM SFFINAL

#J236
```

Example 9-3. Streaming a Job File

Using the STREAM Facility from your Terminal

Although creating a job file with EDITOR or another word processing subsystem is easy and efficient, you can also create the job file and stream it in a single step. This method works well for short, one-time jobs, such as printing the listing produced by a command.

To stream a job from the terminal:

1. Enter :STREAM at the colon prompt.

   The system displays a > prompt.

2. Enter !JOB, followed by the user logon (including any passwords).

3. Enter the commands you want, preceded by the exclamation mark (!) (unless they are subsystem commands).

4. Enter !EOJ

5. Enter a colon (:) character to exit the STREAM facility.

   Example 9-4 shows how to enter the job illustrated in Example 9-2 without first creating a job file on disc.

```
:STREAM

>!JOB JSFFINAL,MAC/SUPER.TECHPUB; RESTART; OUTCLASS=LP,,5
>!RUN SORT.PUB.SYS
>INPUT SFLIST
>OUPUT SFFINAL
>KEY 1, 10
>END
>EXIT
>!EDITOR
>T SFFINAL
>L ALL
>EXIT
>!EOJ
  #J345

> :
:
```

Example 9-4. Streaming a Job from the Terminal

Monitoring a Job

Although you could use the :SHOWIN and :SHOWOUT commands to check on your job, the :SHOWJOB command is by far the most useful.

The :SHOWJOB Command

The :SHOWJOB command displays information about jobs and sessions. You can request information about all jobs and sessions or specify a single job.

When entered without parameters, the :SHOWJOB command displays information about all jobs and sessions on the system, as shown in Example 9-5. Each column shows a particular piece of information. As you read the descriptions of each column, refer to this display.

```
:SHOWJOB

JOBNUM        STATE  IPRI    JIN   JLIST  INTRODUCED  JOB NAME

#S202         EXEC          20    20     MON 11;38A  CONSOLE.SYS
#S367         EXEC          142   142    TUE  6:14A  JAN,ART.SYS
#S540         EXEC   QUIET  35    35     TUE  9:20A  MARY.JONES
#J216         EXEC          10S   LP     TUE 10:30A  MAIL,Y,OFFICE
#J217         WAIT          10S   LP     TUE 12:25A  STATS.ACCT
#S582         INTRO          55   55     TUE  1:02A  PAT.MGR

6 JOBS:
   1 INTRO
   1 WAIT
   4 EXEC; INC 3 SESSIONS
   0 SUSP
jobfence=6; JLIMIT=11; SLIMIT=50

CURRENT:  7/26/87 13:02

JOBNUM    STATE  IPRI    JIN   JLIST  SCHEDULED-INTRO   JOB NAME

#435      SCHED   5      10S   LP     7/26/87    14:21  JOBCHECK/RSPOOL.SYS
#394      SCHED   8      10S   LP     7/26/87    20:00  SWING4,OPERATOR.SYS

2 SCHEDULED JOBS(S)
```

Example 9-5. Sample :SHOWJOB Display, without Parameters

JOBNUM Column

Lists the number assigned by the system to sessions and jobs. The numbers that follow are assigned in the order that people log on or submit jobs and continue until there is a COOLSTART. At that point, session and job numbers start from 1 again.

STATE Column

Shows the session or job's processing state. There are five possibilities for jobs:

- INTRO means the job is being submitted.
- WAIT means the job is waiting because of lack of system resources or high jobfence setting.
- EXEC means the job is starting up.
- SUSP means the job was executing, but is now suspended.
- SCHED means the job is scheduled to execute at the time specified by user in the :STREAM command.

IPRI (Inpriority) Column

Displays the input priority for scheduled jobs. In Example 9-5, for example, job number #J435 has an input priority of 5, which is relatively low. Job number #J394 has an input priority of 8. The exact meaning of these priorities can differ from system to system.

JIN and JLIST Columns

Show the device numbers assigned to input devices (JIN) and output devices (JLIST). Note that with jobs, these numbers differ because jobs are usually input from disc (which is generally ldev 10) and printed (output) on the line printer (LP), unless you specify otherwise.

INTRODUCED Column

This column shows the date and time the job was introduced with the `:STREAM` command.

JOBNAME Column

Shows the user and account name, as well as the job name (if any).

Summary Report

The summary report gives an overview of what is happening on the system. It shows how many jobs and sessions are on the system, and how many are in each state. It also shows the settings for the jobfence the JLIMIT, and the SLIMIT. (If your job seems to be "waiting" for a long time, check the JLIMIT setting to make sure your input priority is higher than the jobfence.) Beneath this information, the summary report lists the current date and time, as well as any sheduled jobs.

NOTE

Since the operations staff usually has good reasons for raising the jobfence to keep users off the system, don't change your priorities without first checking with them.

Example 9-6 shows how to display status information about job number #J236.

```
:SHOWJOB  #S236

CURRENT: 7/26/87 13:32

JOBNUM     STATE   IPRI    JIN     JLIST   SCHEDULED-INTRO   JOB NAME

#S236      SCHED    8      10S      LP      7/26/87   20:00   SWING4,Operator.SYS

jobfence=6; JLIMIT=11; SLIMIT=50
```

Example 9-6. :SHOWJOB Command, with Job Number Parameter

Security of Batch Data

As you know, a job file must include any passwords that apply to the group, user, or account. This presents a security problem, because anyone who can access the job file can read the passwords.

Some companies cope with this problem by creating special programs that insert passwords into job streams so that passwords do not have to be stored with the job file. Check with your manager to find out whether your site uses such a procedure.

NOTE

The HP Security Monitor (HP30392A) is a product that may be purchased separately. It helps solve security problems with the following features:

- Rejects embedded passwords.
- Prohibits users from streaming other users' jobs unless they have special capabilities.
- Allows users to stream their own jobs without having to supply passwords.

You can also use the :ALTSEC command to keep other users from looking at your job files. In this way, others can execute the job, but they cannot see the contents of the job file. For a more detailed explanation, including a summary of access permissions (Table 6-1) and user codes (Table 6-2), refer to Chapter 6 "Basic Account Structure."

To prevent others from looking at a job file:

Enter ALTSEC, followed by the file name and the access codes that specify execute access only to all account members.

```
:ALTSEC filename; (X:AC)
```

Example 9-7 shows how to secure the data in the job file OLLIE by assigning only the EXECUTE access code to other users. Only the file's creator can actually look at the file.

```
:ALTSEC OLLIE (X:ANY;R:CR)
:
```

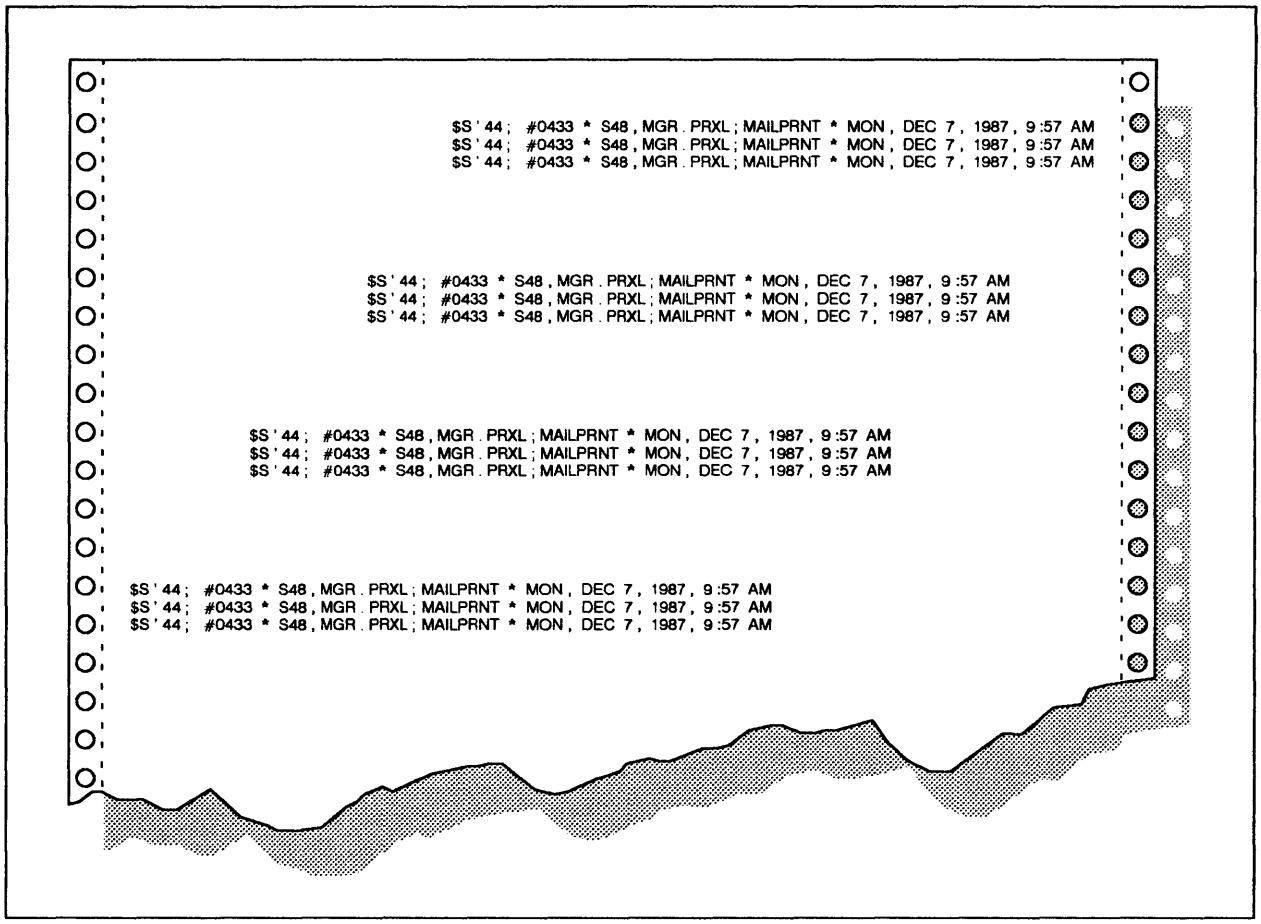
Example 9-7. Securing a Job File with the :ALTSEC Command

The Job Printout

The printout produced by a job consists of three main parts: the header, the main body, and the trailer (tear sheet).

The header and trailer pages mark the beginning and end of each printout. They also serve to separate printouts from each other and make them easy to identify. An example of a header or trailer page is shown in Figure 9-5. Both pages contain essentially the same information, which includes:

- The system-assigned job number.
- The system-assigned DFID (Device File Identification Number).
- The user and account names.
- The output device name, which is usually $STDLIST, or the formal file designator.
- The date and time the job was introduced (header page) and the date and time it finished printing (trailer).



LG200077_061

Figure 9-5. The Trailer Page on the Job Printout

The main body of the printout contains whatever output was generated by the commands in the job.

A

"Breakable" Commands

"Breakable" means a command's execution can be interrupted by pressing the Break key.

A program command invokes a subsystem or runs a user program. All program commands are breakable. You can resume the interrupted program command with the :RESUME command. Table A-1 shows the program commands.

A nonprogram command executes a system command. You cannot resume interrupted non-program commands. Table A-2 shows the nonprogram commands.

Table A-1. Program Commands (All Are Breakable)

:BASIC	:EDITOR	:PREPRUN
:BASICGO	:FCOPY	:RESTORE
:BASICOMP	:FORTGO	:RJE
:BASICPREP	:FORTPREP	:RPG
:COBOL	:FORTRAN	:RPGGO
:COBOLGO	:IMF	:RPGPREP
:COBOLPREP	:MRJE	:RUN
:COBOLII	:PASCAL	:SEGMENTER
:COBOLIIGO	:PASCALGO	:SPL
:COBOLIIPREP	:PASCALPREP	:SPLGO
:DSCOPY	:PREP	:SPLPREP
		:STORE

Table A-2. Nonprogram Commands

BREAKABLE	NONBREAKABLE	
:( ) COMMAND LOGON	:ABORT	:LISTLOG
:HELLO (If there is no OPTION LOGON UDC*)	:ALTLOG	:MOUNT
:HELP	:ALTSEC	:PTAPE
:LISTF	:ASSOCIATE	:PURGE
:LISTVS	:BUILD	:RECALL
:REDO	:COMMENT	:RELEASE
:REPORT	:CONSOLE	:RELLOG
:SHOWCATALOG	:CONTINUE	:REMOTE
:SHOWDEV	:DATA	:RENAME
:SHOWIN	:SET	:RESET
:SHOWJCW	:DEBUG	:RESETDUMP
:SHOWJOB	:DISASSOCIATE	:RESUME
:SHOWME	:DISMOUNT	:SAVE
:SHOWOUT	:DSLINE	:SECURE
:STREAM	:DSTAT	:SETCATALOG
	:ELSE	:SETDUMP
	:ENDIF	:SETJCW
	:EOD	:SETMSG
	:EOJ	:SHOWALLOW
	:FILE	:SHOWCACHE
	:FREERIN	:SHOWLOGSTATUS
	:GETLOG	:SHOWTIME
	:GETRIN	:SPEED
	:IF	:TELL
	:JOB	:TELLOP
		:VSUSER

B

DSCOPY Procedures

The DSCOPY utility is part of the DS/3000 Data Communication Subsystem. It lets you copy files between groups and accounts on different computers, without needing to mount a tape or getting an operator involved in the process. Note that you need a valid logon, including any passwords, for each system.

To copy files between systems:

1. Connect to the system to which you want to copy. (Ask your manager for instructions, if necessary.)
2. Log onto the system to which you want to copy.
3. Connect with the system from which you want to copy.

   ```
   :DSLINE  othercomputer name
   ```

4. Enter the REMOTE HELLO command to log onto the system from which you want to copy.

   ```
   :REMOTE HELLO logon
   ```

5. Copy the file into your system.

   ```
   :DSCOPY filename, othercomputer name
   ```

   The system displays NEW FILE CREATED, followed by the filename, the group, and the account name, and SUCCEEDED

6. Repeat step 4 to copy any other files into this account.
7. (Optional) Check to make sure the file was copied.

   ```
   :LISTF
   ```

8. Return to the MPE operating system.

   ```
   :EXIT
   ```

9. Log off the system you copied from.

   ```
   :REMOTE BYE
   ```

NOTE

The DSCOPY subsystem will not copy the file if a file with the same name already exists.

Index

A

:ABORT, 3-20, 8-15

/ADD, 5-6

/ALTER, 5-40

:ALTSEC, 6-15, 9-16

/APPEND, 5-40

Aborting Programs, 8-15

Access Permissions, 6-7
  corresponding user codes, 6-13
  restricting, 6-16

Account Manager, 6-2

Account Structure, 6-2
  billing, 6-6
  organization, 6-6

Ampersand (&) Character, 3-7

Anatomy
  of a command, 3-3
  of a job file, 9-9
  of a UDC, 4-3

APPEND Access Permission, 6-13

Auto LF, 2-6

B

:BUILD, 3-19, 7-10

:BYE, 3-20, 8-9

Backreferences, 7-18

Backup, 5-27

Batch Processing, 1-10, 9-2

Batch Processing Commands, 9-3

Baud Rate, 2-7

Beep, 2-18

Bell, 2-18

Billing, 6-6

Binary Numbers, 1-8

Block Mode, 2-6

Blocking Factor, 7-10

Blocking Messages, 8-27

BREAK/NOBREAK, 4-4

BREAK/RESET Key, 8-13

Braces ({ }) Character, 3-10

Brackets ([ ]) Character, 3-10

C

!COMMENT, 9-3, 9-7

!CONTINUE, 9-3, 9-8

/COPY, 5-6

Capabilities, 6-7, 6-17
  default, 6-18

Changing
  lockwords, 6-11
  passwords, 6-9

Checking
  capabilities, 6-19
  file access permissions, 6-14
  file equations, 7-19

Clearing, file equations, 7-20

Clearing the Screen Display, 2-18

Colon (:) Character, 3-3

Combining Sorted Files, 5-26

Command Interpreter, 3-2

Command Logon Option, 8-9

Command Sequences, 2-16

Command Sequences Summary, 2-17

Commands
  general information, 3-2
  name, 3-3
  syntax, 3-2

Comment Lines, 4-5

Comparing File Contents, 5-38

Compilers, 1-8

Index (Continued)

Computer Cable Connection, 2-3

Configuration Settings, 2-9

Console, 1-5

Continuation Character, 3-7

Copying
- files, 7-8
- files between accounts, 5-36
- files to nonsharable devices, 5-38
- lines within a file, 5-14

Correcting Commands, 8-10, 8-11

CPU, 1-4

Creating
- a job file, 9-10
- device files, 7-17
- files with :BUILD, 7-11
- files with EDITOR, 5-6
- lockwords, 6-10

CTRL key, 2-16

Cursor Control Keys, 2-15

D

/DELETE, 5-6

Databases, 1-9

Defaults, 1-11, 3-4

Deleting
- file equations, 7-20
- files, 7-7
- spool files, 5-44

Delimiters, 3-6, 3-11

Determining System Access, 8-7

Device Files, using to reroute output, 7-17

Device files, definition, 7-12

Devices, 1-4

Directory, 6-5

Disc Drives, 1-5

Displaying, 8-16
- command example, 3-14
- command operation, 3-14
- command parameters, 3-14
- command syntax, 3-14
- current date and time, 8-16
- existing file equations, 7-19
- file access permissions, 6-14
- file characteristics, 7-21, 7-22
- file contents, 7-5
- information about current session, 8-17
- information about jobs and sessions, 8-20, 9-13
- information about spoolfiles, 5-41, 8-22
- information about temporary files, 7-23
- information aobut peripherals, 8-18
- list of files, 7-4
- sorted files, 5-25
- UDCs, 4-16
- user capabilities, 6-19

Documentation Conventions, 3-10

Documentation Summary, 3-8

DSCOPY, 5-1

E

/EDITOR, 5-6

/EXIT, 5-6, 5-40

Echo, 4-4

EDITOR, 1-9
- copying lines within files, 5-14
- creating files to be sorted, 5-22
- creating new files, 5-6
- deleting lines within files, 5-16
- general information, 5-4
- Help Facility, 5-20
- inserting text, 5-11
- line numbers, 5-4
- modifying existing files, 5-9
- moving lines within file, 5-13
- printing files, 5-19
- scratch files, 5-5

Index (Continued)

setting tabs, 5-22
subcommand summary, 5-6
Edit Control Keys, 2-15
Enq/Ack, 2-7
Error Messages, 3-2
ESC key, 2-16
EXECUTE Access Permission, 6-13
Execution Options
SORT-MERGE, 5-30
UDCs, 4-4
Extents, 7-10

F

:FCOPY, 5-35
FCOPY Subsystem, 7-20
comparing file contents, 5-38
copying files between accounts, 5-36
copying files to nonsharable devices, 5-38
general information, 5-34
messages, 5-35
printing, 5-38
summary of functions, 5-34
using the, 5-35
FCOPY/V, 1-9
Fields, 5-21
File Access Permissions, 6-7, 6-11
displaying, 6-14
File Characteristics, displaying, 7-21
File Code, 7-10
File Commands, 3-18
File characteristics, 7-9
File Equations, 5-29, 5-33, 7-15, 7-16
File equations, 7-19
File System, 7-9
Files
access permissions, 6-7
backreferencing, 7-18
checking exisiting file equations, 7-19
clearing existing file equations, 7-20
copying, 7-8
creating with :BUILD, 7-11
creating with EDITOR, 5-6
default characteristics, 7-10
definition of, 7-2
device files, 7-12, 7-17
disc files, 7-3
displaying contents, 7-5
displaying file characteristics, 7-22
displaying list of, 7-4
finding, 7-4
linking actual files with generic file names, 7-16
lockwords, 6-7, 6-10
modifying with EDITOR, 5-9
naming files, 6-4
organizing, 6-2
printing, 7-20
releasing, 6-16
renaming, 7-6
saving temporary files, 7-24
securing, 6-17
system defined files, 7-12
temporary, 7-23
Finding Commands, 3-13
Formal File Designators, *7-14*
Full Screen Mode, 2-15
Fully Qualified File Name, 6-4
Function Keys, 2-18, 4-2
general information, 2-7

G

/GATHER, 5-6
Groups, 6-2

H

:HELLO, 3-8, 3-20, 8-2
:HELP, 3-20

Index (Continued)

Hard Reset, 8-14

Hardware Components
  CPU, 1-4
  general description, 1-4
  terminals, 1-5

HELP/NOHELP, 4-5

Help Facility
  displaying command operation, 3-14
  displaying command syntax, 3-14
  displaying example, 3-14
  displaying parameters, 3-14
  displaying UDC information, 4-17
  finding commands, 3-13
  general information, 3-12
  SPOOK subsytem, 5-45
  using EDITOR's Help Facility, 5-20

HIPRI Parameter, 8-5, 9-6

HP Security Monitor Product, 9-5, 9-16

I

"Interactive" Jobs, 9-12

INPRI= Parameter, 8-5, 9-6

Information Management, HP systems, 1-9

Input Device, 2-2

Input Priority, 8-7

Interactive Processing, 1-10

Interrupting
  a subsystem, 2-18
  commands, 8-13
  communications with the computer, 2-14
  programs, 8-15
  screen listings, 2-17
  subsystems, 8-15

Intrinsics, 7-9

J

:JOB, 9-3

JLIMIT, 8-7

Job Name, 9-5

Job Printout, 9-17

Job Prompt, 9-3

Jobfence, 8-5, 8-7

Jobs, 1-10

K

/KEEP, 5-6

Keyboard, 2-11
  connection to computer, 2-5
  unlocking, 8-14

Keys
  cursor control, 2-15
  edit control, 2-15
  function keys, 2-18
  terminal control, 2-14

Keyword Parameters, 3-5, 3-10

KSAM files, deleting, 7-8

KSAM/V, 1-9

L

/LIST, 5-6, 5-40

:LISTEQ, 3-19, 7-19

:LISTF, 3-19, 7-4, 7-21

:LISTFTEMP, 3-19, 7-23

:LISTUSER, 6-19

LDEV Number, 1-5

LIST/NOLIST, 4-4

LISTDIR5, 6-14, 6-19, 7-21, 7-22

Line Numbers, 5-4

Listing, 2-23

LOCK Access Permission, 6-13

LOGON/NOLOGON, 4-5

Local Mode, 2-6

Index (Continued)

Lockwords, 6-7, 6-10
Logging Off, 8-9
Logging On, 8-2, 8-6
Logon UDC, 4-5

M

/MODIFY, 5-6
Mainframe Computers, 1-3
Manual Conventions, 3-10
MERGE Program, 5-26
Merging Sorted Files, 5-26
Message Commands, 3-21
Microcomputer, 1-2
Minicomputer, 1-3
Modem Connection, 2-4
Modifying Files, 5-9
Monitoring Commands, 3-17
Monitoring Jobs, 9-13
Monitoring Sessions, 8-16
Moving
  lines within a file, 5-13
  the cursor, 2-15
Multiprocessing, 1-3
Multiuser System, 1-3

N

$NEWPASS, 7-13
$NULL System Defined File, 7-13
Nested UDCs, 4-5, 4-8

O

$OLDPASS System Defined File, 7-13
Operating System, general information, 1-9
Optional Parameters, 3-4, 3-10
OUTCLASS Parameter, 9-7
Outfence, 8-5, 8-7
Output Device, 2-2
Output Priority, 8-8, 9-7

P

:PURGE, 3-19, 7-7
/PURGE, 5-40
Parameters
  delimiters, 3-11
  general information, 3-4
  keyword, 3-5, 3-10
  optional, 3-4, 3-10
  positional, 3-5, 3-10
  required, 3-4, 3-9, 3-10
  specifying values, 3-10
  subparameters, 3-5, 3-10
  summary of characteristics, 3-6
  UDCs, 4-4
Parity, 2-7
Passwords, 6-7, 8-3, 9-4, 9-16
PC, 1-2
Peripherals, 1-4
Positional Parameters, 3-5, 3-10
Power Cable Connection, 2-2
PRI= Parameter, 8-5, 9-6
Printer Cable Connection, 2-21
Printers, 1-7
Printing
  files with EDITOR, 5-19
  files with FCOPY, 5-38
  listings on system printer, 2-23
  Screen Dump at Workstation, 2-22
Printing the Screen Display, 2-21
Programming Function Keys, 2-19
Prompt, 3-3
Purging Files, 7-7

Index (Continued)

Q

QUERY/5, 1-9
Quiet Mode, 8-27

R

:REDO, 3-20, 8-10
:REDO Subcommands, 8-10
:RELEASE, 5-35, 6-15, 6-17
:RENAME, 3-19, 6-10, 7-6
:REPORT, 3-18
:RESET, 3-19, 7-16, 7-20
:RESTORE, 5-28, 5-32
:RESUME, 8-15
:RUN, 3-20, 5-3, 8-13
READ Access Permission, 6-13
RESTART Parameter, 9-7
Record Format, 5-21
Record Size, 7-10
Record Structure, 7-10
Record Type, 7-10
Records, 5-21
Releasing Files, 6-16
Remote Mode, 2-6
Renaming files, 7-6
Repeating Commands, 8-10, 8-12
Required Parameters, 3-4, 3-9, 3-10
Rerouting Output, 7-17
Resetting the Terminal, 2-14
Restricting File Access Permissions, 6-16
Resuming Screen Listings, 2-17
Retrieving Lockwords, 6-11
Runnning Programs, 8-13

S

:SAVE, 3-19, 7-24
:SECURE, 6-15
:SETCATALOG, 3-20, 4-6, 4-10
:SETMSG, 3-20
:SETMSG OFF, 8-27
:SETMSG ON, 8-27
/SHOW, 5-40
:SHOWALLOW, 3-18, 8-16
:SHOWCATALOG, 4-16
:SHOWDATALOG, 3-20
:SHOWDEV, 3-18, 8-18
:SHOWJOB, 3-18, 8-20, 9-13
:SHOWME, 3-18, 8-17
:SHOWOUT, 8-22
:SHOWTIME, 3-18, 8-16
:SORT, 5-5
/SPOOK, 5-40
$STDIN System Defined File, 7-13
$STDLIST System Defined File, 7-13
:STORE, 5-28, 5-29
:STREAM, 9-3
SAVE Access Permissions, 6-13
Saving Temporary Files, 7-24
Scratch Files, 5-5
Screen Buffer, 2-11
Screen Dumps, 2-21
Screen Echo, 2-17
Scrolling, 2-11
Securing files, 6-17
Security
  of batch data, 9-5, 9-16
  overview of provisions, 6-7
Sending and Receiving Messages, 8-25

Index (Continued)

Sending Messages to Operator, 8-26

Sending Messages to Users, 8-26

Session Name, 8-4

Sessions, 1-10

SLIMIT, 8-7

SORT-MERGE Subsystem, 1-9
  general information, 5-20
  SHOW parameter, 5-30
  SORT Program, 5-23

SORT-MERGE Subsytem, required record format, 5-21

Soft Reset, 8-14

Software Components
  FOS, 1-8
  general description, 1-4
  overview of included software, 1-8

Sorting Files, 5-20, 5-23

SPOOK Subsystem
  adding information to spool files, 5-43
  changing printer, number of copies, 5-44
  deleting spool files, 5-44
  displaying spoolfiles, 5-41
  general information, 5-39
  Help Facility, 5-45
  locating data in spool files, 5-42
  summary of subcommands, 5-40

SPOOK Tapes, 5-39

Spool Files, 5-39

SQL, 1-9

STORE-RESTORE Subsystem, 1-9
  general information, 5-27
  PROGRESS parameter, 5-32
  PURGE parameter, 5-31

State of Spool File, 5-41

Stopping Program Execution, 8-15

Streaming a Job, 9-12

Streaming a Job from the Terminal, 9-12

Submitting a Job File, 9-12

Subparameters, 3-5, 3-10

Subsystems, 1-9
  associated formal file designators, 7-14
  EDITOR, 5-4
  FCOPY, 5-34
  general information, 5-2
  interrupting, 8-15
  SORT-MERGE, 5-20
  SPOOK, 5-39
  STORE-RESTORE, 5-27
  summary of, 5-3

Syntax, 3-2

System Console, 1-5

System Defined Files, 7-12

System Manager, 6-17

T

!TELL, 9-3

:TELL, 3-21, 8-26, 9-7

:TELLOP, 3-21, 8-26, 9-3, 9-7

/TEXT, 5-6, 5-40

Tabs, 5-22

Tape, 5-28

Tape Drives, 1-6

TERM= Parameter, 8-5

Tear Sheet, 9-17

Terminal Control Keys, 2-14

Terminals, 1-5, 2-2, 2-18
  changing settings, 2-7
  general information, 2-2
  keyboard, 2-11
  resetting, 2-14, 8-14
  settings, 2-2, 2-5
  summary of settings, 2-6
  using PCs as, 1-2

TIME= Parameter, 8-5, 9-6

TurboIMAGE, 1-9

Turning Messages Off, 8-27

Index (Continued)

Turning Off Screen Echo, 2-17
Turning on Messages, 8-27

U

UDC File, 4-6
UDCs
  adding, modifying, and removing, 4-10
  body, 4-5
  comment lines, 4-5
  comparison to function keys, 4-2
  construction, 4-3
  displaying, 4-16
  execution options, 4-4
  general information, 4-2
  levels, 4-2
  nested, 4-5, 4-8
Unlocking the Keyboard, 8-14

V

VPLUS, 1-9

W

Warning Message, 8-25
Welcome Message, 8-6, 8-25
Wild Card Characters, 7-5
WRITE Access Permission, 6-13
Writing File Equations, 7-15, 7-16

X

/XPLAIN, 5-6, 5-40

READER COMMENT SHEET

Information Technology Group

General User's Reference Manual

Manual Part Number 32033-90158 October 1988

A reader comment sheet helps us to improve the readability and accuracy of the document. It is also a vehicle for recommending enhancements to the product or manual. Please use it to suggest improvements.

SERIOUS ERRORS, such as technical inaccuracies that may render a program or a hardware device inoperative should be reported to your HP Response Center or directly to a Support Engineer. An engineer will enter the problem on HP's STARS (Software Tracking and Reporting System). This will ensure that critical and serious problems receive appropriate attention as soon as possible.

Editorial suggestions (please include page numbers):_____

_____

_____

_____

Recommended improvements (attach additional information, if needed):_____

_____

_____

_____

Name _____ Date _____

Job Title _____ Phone_____

Company_____

Address _____

HP 1000 Series———————— (e.g., E-series, A400, A600, etc.)
HP 3000 Series———————— (e.g., 37, 68, 930, etc.)
HP 9000 Series———————— (e.g., 300, 840, etc.)

☐ Check here if you would like a reply.

Hewlett-Packard has the right to use submitted suggestions without obligation, with all such ideas becoming property of Hewlett-Packard.

Fold Here

NO POSTAGE NECESSARY IF MAILED IN THE UNITED SATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1070, CUPERTINO, CA 95014

- POSTAGE WILL BE PAID BY -

Publications Manager
Hewlett-Packard Company
System Technology Division
19483 Pruneridge Avenue
Cupertino, California 95014-9974

Fold Here



Part Number 32033-90158
Edition 1
E1088
Printed in U.S.A.