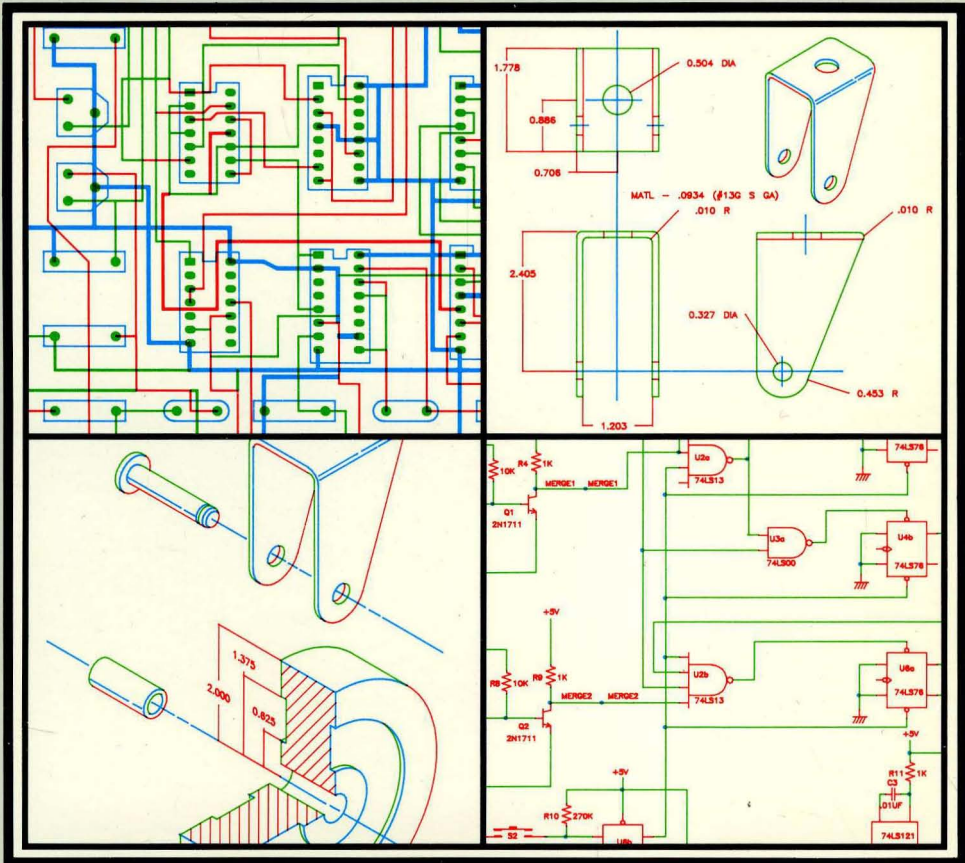


Managing and Editing Files with HP EGS 2.1



HP EGS Manuals

Installing HP EGS lists the hardware and software needed for an HP EGS system, gives some hints on connecting the hardware, and explains installing the HP EGS software on a hard disc and Shared Resource Manager. Appendices contain information on using a hard disc and Shared Resource Manager, the differences between HP EGS 2.0 and 2.1, and combining HP EGS 2.1 with a complete Pascal 3.1 Operating System.

Learning HP EGS provides tutorials to familiarize you with the HP EGS system and its personalities: General Drawing, Mechanical Drafting, Electrical Schematic Drawing, and Printed Circuit Board Layout. By following the exercises in this manual you will learn the basic concepts of HP EGS, so that you can do more advanced tasks easily.

Understanding HP EGS describes the theory and operation of the HP EGS system. It explains how the Graphics Editor works and how to customize the system. You can learn how to use HP EGS post-processors to create material and connection lists, produce output to run photoplotter and drill machines, and translate drawings to and from the IGES Translator. An application note on using archive files to transport data is also included.

Managing and Editing Files with HP EGS describes file system concepts and provides tutorial exercises to help you learn how to use the text Editor and Filer. A syntax reference further explains the Editor, Filer, and Pascal Command Line commands.

HP EGS Syntax Reference lists all HP EGS commands. Each command is illustrated with a diagram, explained in a table, and shown in use with one or more examples. The introductory sections of the reference explain how to interpret the diagrams and enter the commands. The commands used in archive files are listed at the end of the reference.

Managing and Editing Files with HP EGS 2.1 *for the HP 9000 Series 200/300 Computers*

Manual Set Reorder No. 98305-90008

© Copyright 1985, Hewlett-Packard Company.

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

Use of this manual and flexible disc(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs can be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the Rights in Technical Data and Software clause in DAR 7-104.9(a).



Hewlett-Packard Company
3404 East Harmony Road, Fort Collins, Colorado 80525

Printing History

New editions of this manual will incorporate all material updated since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revision date at the bottom of the page. A vertical bar in the margin indicates the changes on each page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

July 1985...Edition 1

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 MANUAL, 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 direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

WARRANTY

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Table of Contents

Chapter 1: Files, Volumes, and Mass Storage

Introduction	1
Understanding the File System	1
Files	1
Volumes	3
Units	5
The System Volume and Default Volume	5
Volume Space	7
The Shared Resource Management System	8
Hierarchical Directories	8
Syntax of a Directory Path	9
Notation	10
SRM Units and Volumes	10
Moving Up and Down the Hierarchy	11
Creating a Directory	12
System Utilities for File Manipulation	13
Initializing Flexible Discs	13
Large File Management	15
9-Track Magnetic Tape and Paper Tape	16
Character Code Converter	16
Reconfigure System	17
HP EGS Files and Volumes	18

Chapter 2: Using the Text Editor

When to Use the Editor	27
Entering the Editor	28
Adding Text (Insrt)	29
Confirming or Aborting Commands	31
Refreshing the Screen	31
Moving the Cursor	32
Deleting and Replacing Text	34
Deleting Text (Dlete)	34
Deleting Large Sections of Text (Zap)	35
Typing Over Characters (Xchng)	36
Replacing Character Strings (Rplace)	37

Copying and Moving Text (Cpy)	40
Searching for Text Patterns (Find)	42
Formatting Text	43
Defining the Format (Set environment)	43
Adjusting Margins (Adjst)	45
Filling Paragraphs (Margin)	46
Printing a File	47
Exiting the Editor	48
Finding More Information.....	49

Chapter 3: Using The Filer

Entering the Filer	52
Manipulating Volumes and Directories	53
Listing Volumes (Vols)	53
Changing the Default Volume (Prefix-vol)	54
Listing Directories (Ldir)	55
More Directory Information (Ext-dir)	58
Creating Files	59
Manipulating Files	59
Copying and Moving Files (Filecopy)	59
Changing File Names (Change)	60
Removing Files from a Volume (Remove)	61
Changing File Types and Printing Files (Translate)	61
Introducing Workfiles	63
Copying a File into a Workfile (Get)	63
Copying a Workfile Back to the Original File (Save)	65
Clearing a Workfile (New)	66
Displaying the Workfile Name and State (What)	66
Managing Local Mass Storage	67
Checking Mass Storage Readability (Bad secs)	67
Compacting Files in a Volume (Krunch)	67
Creating an Empty Directory (Zero)	68
Using the SRM File Commands	69
Using Passwords to Protect Files (Access)	69
Making Directories and Files (Make)	69
Making New Pointers to a File (Duplicate link)	69
Changing the Working Directory (Udir)	69
Exiting the Filer	70
Finding More Information.....	70

Chapter 4: Pascal Command Line

Information Provided	71
The Pascal Command Line	72
Pascal Command Quick Reference	73
Hpegs	74
Initialize	75
Memvol	76
Newsysvol	78
Permanent	79
Run	80
Stream	81
User restart	84
Version	85
What	87
eXecute	89

Chapter 5: Editor Reference

Information Provided	91
Editor Command Quick Reference	92
Adjust	93
Copy	95
Delete	97
Find	99
Insert	102
Jump	105
Margin	106
Page	107
Quit	108
Replace	111
Set	115
Verify	120
eXchange	121
Zap	123

Chapter 6: Filer Reference

Information Provided	125
Filer Command Quick Reference	126
File and Volume Specifications	127
Access	128
Bad sector	130
Change	131
Duplicate link	133
Extended directory	135
Filecopy	138
Get	141
Krunch	142
List directory	144
Make	146
New	149
Prefix	150
Quit	152
Remove	153
Save	155
Translate	157
Unit directory	159
Volumes	161
What	162
Zero	163

Appendix A: Flexible Disc Handling Guidelines

Inserting a Disc Into Your Disc Drive	167
Inserting a Micro Disc	167
Inserting a Manual Shutter Disc	168
Removing a Micro Disc	168
Inserting a Mini Disc	169
Removing a Mini Disc	169

Appendix B: Run-Time Errors

Files, Volumes, and Mass Storage

Chapter 1: Files, Volumes, and Mass Storage

Introduction	1
Understanding the File System	1
Files	1
Volumes	3
Units	5
The System Volume and Default Volume	5
Volume Space	7
The Shared Resource Management System	8
Hierarchical Directories	8
Syntax of a Directory Path	9
Notation	10
SRM Units and Volumes	10
Moving Up and Down the Hierarchy	11
Creating a Directory	12
System Utilities for File Manipulation	13
Initializing Flexible Discs	13
Large File Management	15
9-Track Magnetic Tape and Paper Tape	16
Character Code Converter	16
Reconfigure System	17
HP EGS Files and Volumes	18

Files, Volumes, and Mass Storage

Chapter

1

Introduction

This chapter discusses files and volumes, their use on local mass storage devices, the Shared Resource Management (SRM) system, and the HP EGS file utility programs. In addition, it lists the HP EGS files contained in each system volume, excluding only the names of the drawing files (also known as library parts).

Understanding the File System

The most elemental packet of information is the file.

Files

Think of a file as a file folder that holds information. The information in a file might be a drawing, a parts list, or a memo.

File folders are usually labeled so you can distinguish among them, and so are files. Each file has a file name associated with it that distinguishes it from every other file.

The terms “file specification” and “file name” refer to the two ways you may reference a file. A file name is a character string which externally identifies a file in a volume or directory. A file specification is a character string which consists of the file name plus several other optional items: volume name, directory path name, and passwords. Not all of these items are allowed under all circumstances; for instance, directory path names and passwords are only used with the hierarchical directory organization of the SRM.

File Specifications

A file specification may appear in one of two forms. The first form consists of an optional volume ID ending in a colon, an optional directory path, and a required file name (#5:/MY_SRM/WORK/my_file). The second form consists of a volume ID followed by a file name (MYVOL:my_file). The first form is used with SRM systems, the second with local discs.

2 Files, Volumes, and Mass Storage

File Names

When you name files use the following conventions:

- File names should begin with an alphabetic character.
- File names should be restricted to alphanumeric characters and the underscore (_).
- File names can be up to eight characters long, not including extensions (for example, _d or _ms)

If you name a file using uppercase letters, the Graphics Editor will convert them to lowercase. If you enter an uppercase file name with a command, the Graphics Editor converts it to lowercase, with four exceptions. The CHANGE, FILECOPY, INPUT, and REMOVE commands do not automatically change uppercase to lowercase, because you may want to perform operations on files which are not drawings.

Because some characters have significance either in Filer commands or in the overall specification of files, you should avoid them. These characters include:

\$? =	Filer wildcard characters.
* : #	Used in specifying volumes.
/ < >	Have special meaning with the SRM.
[]	Used to specify the size of a file when it is created.
, space	HP EGS delimiters.
control characters	Control characters are removed from file names.

Also note that in LIF directories (the standard format for non-SRM volumes) and SRM directories, uppercase and lowercase letters are significant (for example, CHARLIE is not the same file as Charlie).

When saving a file, the system usually appends a suffix to it that denotes its file type. For example, the suffix _d is added to files to indicate that they are drawing files. These suffixes need not be included when you manipulate files within HP EGS, even though they appear in the stored file names.

The following table contains the suffixes used by HP EGS:

_Pr,TEXT	Process file
_mc,TEXT	macro file
_mn,TEXT	screen menu file
_tm,TEXT	tablet menu file
_vs,TEXT	volume search table
_st,TEXT	stroke table
_ms,ASC	message file
_mi,ASC	index to message file
_cn,ASC	control file
_ec,TEXT	echo file
_d	drawing file
_i	macro instance file
_a	archive file
_g	generate file
,ARC	EGS/45 archive file
,GEN	EGS/45 generate file
_c	connection list output
_e	error file from connection listing
_m	material list output
_dbe	material list data base from HP EGS
_Pf	Parts file for material list Post-Processors
_r	rat's nest output
,TEXT	text file
,ASC	ASCII file
,CODE	code file

Volumes

File folders should be stored in a file drawer when not in use. Similarly, files are stored in volumes. A volume is nothing more than a collection of files.

A volume also has a directory associated with it. The directory keeps track of all files in the volume: what their names are, where they are located in mass storage, how big they are, and when they were last modified.

The directory is necessary because, unlike a filing cabinet, HP EGS does not keep its files in alphabetical (or any other) order. It just searches for the first space in the volume big enough to hold the file, and slides it into the slot. If a real file drawer was organized this way, it would be nearly impossible to find anything--unless, of course, it had a directory.

4 Files, Volumes, and Mass Storage

The directory is the organized part of a disorganized volume. When the system stores a file in a volume, an entry is made for it in the directory. Later, when the file is retrieved, the disc drive uses the directory to find the requested location of the file in the volume.

Again using the filing cabinet analogy, a directory would be a large list, posted on the front of a file drawer, showing the names, locations and other information about all files stored in that drawer. The file clerk would keep this list up-to-date and would use it to find a requested file.

Listing Volumes

While you are running the HP EGS Graphics Editor, you may see what volumes are available in your system using the VOLUMES command. If you type in:

```
VOL ; 
```

you will obtain a display like this:

```
Volumes on-line:
 1  CONSOLE:
 2  SYSTEM:
 3  # BOOT:
 6  PRINTER:
11  * SYSVOL:
12  * EWSYS:
13  # EWCODE:
14  # EWEE:
15  # EWME:
16  # EWRC:
17  # EWOPT:
Prefix is - EWSYS:
```

You will get the same result if you use the Volumes command of the Filer.

Naming Volumes

Volumes, like files, must have names so you can tell them apart. A volume name is usually called a volume ID (for IDentifier). A volume ID is like a label on the front of a file drawer that distinguishes it from the other drawers in the filing cabinet. A well-chosen volume ID, like a well-chosen file drawer label, should suggest what is stored inside.

Use the following conventions when naming volumes:

- A volume name should suggest what kind of information is stored in it.
- A volume name can be up to six characters long in local volumes and 16 characters long in SRM directories.
- Each volume name must be followed by a colon (:).

Note

Because the file system works with named volumes, it is important to have only one volume of a given name active at one time. If there are duplicate-named volumes on-line, the file system may destroy data by using one volume when you mean the other.

Some volume names do not correspond to any disc device at all. For example, the file system has a name for each input/output device, like `SYSTEM:` for the keyboard, `CONSOLE:` for the CRT, and `PRINTER:` for the system printer.

Units

Now suppose every filing cabinet in the office is stacked one on top of another to form one tall tower of file drawers. Each file drawer in the tower is assigned a unique number. Actually, it is not the file drawer that is assigned the number, but the location in the tower where the file drawer is kept (that is, the slot that the drawer slides into).

With this scheme, every file drawer is associated with a number that specifies its location in the tower. Since the numbers are attached to file drawer locations rather than to the file drawers themselves, you can remove one file drawer, replace it with another, and the new file drawer will now be associated with that number. Also note that a number may refer to an empty location that has no file drawer.

This situation exists in the HP EGS system. Each volume location in the file system is called a unit. Each unit is assigned a unique unit number when the system is booted. Flexible disc drives can hold only one volume (on flexible disc) and are thus assigned one unit number. But mass storage discs can hold multiple volumes and thus are assigned one unit number for each volume they contain.

To refer to a unit instead of a volume, use `#` followed by the unit number. For instance, `#11:` and `SYSVOL:` would both refer to the same volume in the directory listing previously shown.

The System Volume and Default Volume

Although your workstation can deal with many volumes – up to 50 on-line at once – there are two volumes referred to so frequently that they have special abbreviations. They are the system volume and the default volume.

6 Files, Volumes, and Mass Storage

The System Volume

The system volume is used by the operating system to store its own private files and records. Since the operating system is always overseeing the operation of your computer, the system volume must always be accessible. The abbreviated name for the system volume is * (an asterisk); it appears next to the system volume in the listing you obtain when you enter the VOLUMES command of the Graphics Editor. The asterisk need not be followed by a colon, since it is distinctive. The following notations all denote the system volume:

```
*  
*:  
SYSVOL:
```

Some programs use this system volume for temporary disc storage, so it is wise to leave ample room on the volume for this purpose.

The Default (Prefix) Volume

The other special volume is the default volume, sometimes called the prefix volume. In many HP EGS applications, it is convenient to place frequently needed files together in a single volume. If these files are accessed often, you may find it tedious to constantly type in the volume name or unit number. You can instead tell the system that when no volume name is specified, the one to use is the default volume. You may specify which volume is the default volume by using the PREFIX command of the Graphics Editor. For instance, typing in:

```
PREFIX EWEE: ; 
```

causes the volume named EWEE: to become the default volume. You can also specify a disc drive as the location of the default volume. Typing in:

```
PREFIX #3: ; 
```

causes the right disc drive in an HP Model 236 to become the location of the default volume. The disc installed in it at the time the PREFIX command is entered will be the default volume.

An error will occur if the unit specified with the PREFIX command does not contain a disc (i.e., if you specify #4: and there is no flexible disc in drive #4).

Note that the default volume and the system volume can be the same volume.

Volume Space

Discs have a limited amount of storage capacity. If you try to save a file when there is no room on a volume to create a new file, HP EGS will report that the volume is full. You may be able to solve this problem by using the PACK command of the Graphics Editor, or the Krunch command from the Filer. This command consolidates the free space on a volume by moving all files to the front of the volume.

Note

If possible, keep an empty volume on line at all times in case you run out of storage space during an editing session.

Local disc organization is designed for “contiguous file space allocation.” This means that when space is reserved for a file, the disc areas set aside are sequential. Let’s go back to the analogy with file folders in a drawer. If you had a file too big for one folder, you might put it in two or three folders; of course, you’d want to store them next to each other, not in random places in the drawer.

When a file is purged, all of its space becomes available for use by another file. As files are created and purged, the disc develops “holes” of free space between valid files. This is called “fragmentation.” It’s possible for a considerable amount of free space to exist in the volume, yet be unusable because it is in pieces too small to use. Since files tend to be small compared to the total space on a volume, this problem usually occurs when the volume has relatively little free space left.

To see how fragmented your volume is, use the Extended Directory command of the Filer. This command lists the files on the volume as well as any “holes.”

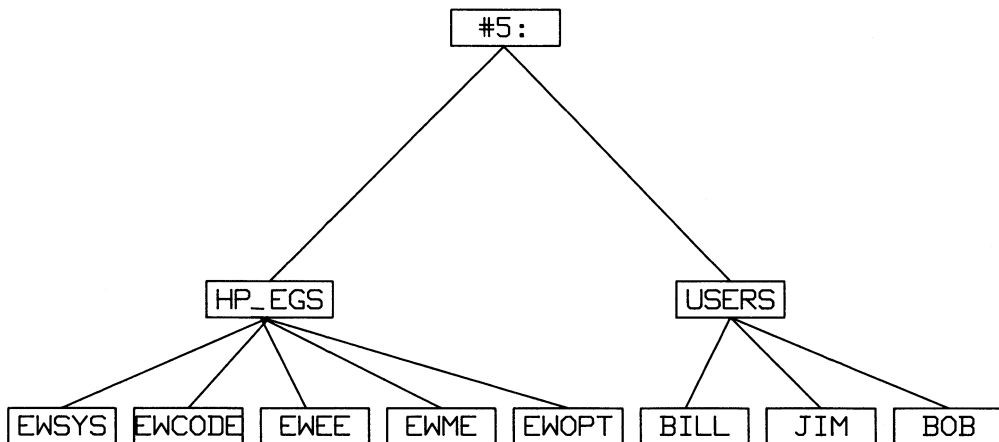
The Shared Resource Management System

The concepts presented so far apply mostly to local mass storage devices. The same concepts also extend naturally to shared mass storage.

The Shared Resource Management System (SRM) allows several workstations (computers) to be connected into a network that allows sharing of files and resources. This network is controlled by a system controller. Since files can be shared between several users, a new directory structure is needed.

Hierarchical Directories

The Shared Resource Management System uses a hierarchical directory structure to organize its files. Thus the first, or top, directory in the structure is called the “root directory.” Subordinate to the root directory are other directories which, in turn, may have further subordinate directories. Each directory may contain either files or other directories. When a directory contains only files, it is called a “leaf directory.” All files can be called “leaf files.” The drawing below shows a hierarchical directory structure:



Each file and directory is uniquely specified by the list of directories from the root to the file. That means several identical file names can exist without confusion if they are in different locations in the directory structure. This is exactly like having several files with the same name in different volumes of a local disc.

Syntax of a Directory Path

Directory paths are only allowed when specifying files on an SRM. There are two special directory names allowed with the SRM. The name `.` (a single period) refers to the current directory somewhere along a path to a file of an SRM logical unit. The name `..` (two periods) refers to the parent of the current directory. Other file names occurring in a directory path are directories along the path to the one which contains the file being specified.

Passwords are sequences of up to 16 characters, which govern the access rights to a file or directory.

Note that a directory path doesn't appear by itself; it appears as part of a file specification, with the file name after the directory path. Examples of directory paths are:

<code>/</code>	Denotes the root directory.
<code>./<PASS1>/</code>	Denotes root, using password "PASS1".
<code>/USERS/ROGER/</code>	Denotes directory ROGER in USERS, which is in the root directory.
<code>HERE/THERE/</code>	Denotes directory THERE, found in HERE
<code>../THERE<PASS2>/</code>	Directory THERE, found in the parent of the current working directory.

A directory path together with a volume ID might appear as

```
#5:/HP_EGS/EWSYS/
```

Because HP EGS has a feature called a volume search table (a `_vs.TEXT` file supplied with each HP EGS personality), you need not specify the complete path name for files that are in volumes or directories listed in the table. The system will automatically search in those volumes for files.

Notation

Hierarchical directories are simple to understand, but they have some specialized words and notation specific to them. For instance, the directory at the top of the hierarchy is called the “root” directory. To refer to a file or directory which is immediately under the root, like HP_EGS in the illustration above, we would write

```
/HP_EGS
```

This is referred to as “slash HP_EGS” or “stroke HP_EGS”. The / indicates the root directory.

To go further down the hierarchy, for instance to EWSYS under HP_EGS, we would write

```
/HP_EGS/EWSYS
```

As you can see, we must specify a file by listing directories from either the root directory or the current working directory to the target file. The list is delimited with a /. Such a sequence of strokes and file names is called a directory path, since it indicates the path one must follow down the hierarchy to get to a particular file.

SRM Units and Volumes

A workstation connected to an SRM normally has units #5: and #45: set up for SRM access. The use of two units is in keeping with the idea that there are usually two special volumes (the system volume and the default volume) through which most file accesses occur.

If the workstation has no local mass storage, unit #45: will automatically be configured by the operating system to be the system volume, and unit number #5: will be available for use as the default volume. If there is local mass storage as well as an SRM connection, the operating system will default to the local disc.

Here is what the HP EGS VOLUMES command might show in an SRM-only configuration:

```
Volumes on-line:
 1  CONSOLE:
 2  SYSTEM:
 3  # BOOT:
 5  # MY_SRM:
 6  PRINTER:
45  * SYSTEM:
Prefix is - MY_SRM:
```

You can see that the system starts out with #5: as the default volume and #45: as the system volume.

Where do the names MY_SRM: and SYSTEM: come from? They are actually the names of particular directories in the SRM's hierarchy. In this example, the name of the SRM volume is MY_SRM. Since there is a directory SYSTEM, it is selected as the system volume. All of this selecting is done by the operating system, as it automatically configures the system each time you boot it.

Moving Up and Down the Hierarchy

It would be tedious to type a directory path every time you wanted to access a file. To avoid this, you can specify the "current working directory" using the PREFIX command of the Graphics Editor. The current working directory can be used as the "root" to specify subordinate files. To use the PREFIX command, type:

```
PREFIX #5:/HP_EGS ; Return
```

Once you have done this, unit #5: is, in effect, a volume named HP_EGS: which contains all the files under the directory HP_EGS in the hierarchy. It's as if you had inserted a disc called HP_EGS: in a disc drive. If you now type the CATALOG command from the Graphics Editor:

```
CAT HP_EGS ; Return
```

it will list all the files in subdirectory HP_EGS. You could also use the command

```
CAT #5 ; Return
```

since the directory HP_EGS was installed in #5 by the PREFIX command.

Suppose that under HP_EGS is another directory named EWMINE which contains more files. To list the files in EWMINE, type:

```
CAT EWMINE ; Return
```

The system will realize that EWMINE under volume HP_EGS is itself a directory, and list its contents. If EWMINE was not a directory, it would simply be listed as a file in directory HP_EGS.

You can move the current working directory still farther down the hierarchy in the obvious way. For instance, to make EWMINE the current directory of #5, you could type:

```
PREFIX #5:/EWMINE ; Return
```

There is no need to specify the entire path name from the root, because EWMINE is already accessible as a file within the volume HP_EGS.

12 Files, Volumes, and Mass Storage

A special notation is provided to move up the hierarchy. Two periods can be used to denote the “parent” directory of a file. For instance, after moving down to EWMINE, unit #5: could be moved back up to the parent directory HP_EGS by:

```
PREFIX .. ; 
```

To go up only two levels, use the double-period twice, separated by a slash:

```
PREFIX ../.. ; 
```

This can be executed all the way up to the root directory. Of course, if you want to get all the way to the top, it is easier to go there directly, using a stroke as the “name” of the root directory. For instance, while #5: is assigned to EWMINE, you could list all the files in the root directory with the command

```
CATALOG #5:/ ; 
```

Creating a Directory

From time to time, you may want to create a new directory on SRM systems. This is quite easy to do. Suppose you want to add a directory named JULY to an existing directory named GROUP. Perform the following operations:

1. Starting from the HP EGS System Utilities menu, press for Filer.
2. Press for Make.
3. Press for Directory.
4. Type in #5:/GROUP/JULY
5. Press to confirm the name you entered.
6. Press to quit the Filer.

You now have a directory named JULY on the directory GROUP.

For more detailed information about creating directories and assigning passwords, please refer to the Filer Reference pages in this manual.

System Utilities for File Manipulation

Several utilities are available from the HP EGS Manager menu to help you manage your files. These utilities allow you to break up and re-combine large files, transfer individual files to and from 9-track magnetic tape and paper tape, and convert character codes.

Refer to the appendix in this manual for flexible disc handling guidelines.

Initializing Flexible Discs

New flexible discs must be initialized (formatted) before they can be used. The program is run by selecting Media Initialization from the System Utilities Menu.

CAUTION

ONLY NEW DISCS SHOULD BE INITIALIZED. IF YOU WANT TO RE-INITIALIZE A DISC FOR ANY REASON, BE SURE TO SAVE ITS CONTENTS ON ANOTHER DISC.

Follow these steps to initialize a flexible disc:

1. Make sure the HP EGS Manager menu appears on the screen:

```
Use arrow keys to move cursor to desired function.  
Select function with <RETURN> or <ENTER>.  
->System Utilities  
    General Drawing Editor  
    Electrical Engineering  
    Mechanical Engineering
```

14 Files, Volumes, and Mass Storage

2. Move the → to System Utilities and press **Return**. The System Utilities menu looks like this:

```
Use arrow keys to move cursor to desired function.  
Select function with <RETURN> or <ENTER>.
```

```
->RETURN To Main Menu  
Media Initialization  
CS80 Tape Backup  
  
Editor  
Filer  
Pascal Command Line  
  
Break-up A Large File  
Re-create A Large File  
Mag/Paper Tape Utility  
Character Code Converter  
Message File Utility  
  
Install Codeword  
Re-boot System  
Reconfigure System
```

3. Move the → to Media Initialization and press **Return**. The following display appears on the screen:

```
Volume ID?
```

4. Insert the flexible disc into disc drive #3 and type:

```
#3: Return
```

You will see the following display:

```
Device: internal minifloppy,0  
Logical unit #3 - ' ' '  
WARNING: the initialization will also destroy:  
#43 <no dir>
```

Press **Y**.

5. The following prompt may or may not appear:

```
Formatting option? (defaults to 0)
```

If it does, refer to your disc drive manual for instructions on selecting the correct formatting option.

6. Prompt:

```
Interleave factor? [1,,15] (defaults to 1)
```

(Your prompt may contain different values.)

Press **Return**

7. Wait about three minutes for the message:

```
Volume zeroing completed.
```

The display immediately returns to the system utilities menu.

8. Remove the initialized disc.

Large File Management

There are some cases in the HP EGS system when large files are created – specifically archive files, generate files, and plot files. It may become necessary to transport a large file via a medium (say a flexible disc) that is too small to contain the entire file. For this problem, there are two HP EGS routines you may access: `BREAK-UP A Large File` which breaks-up large files into a series of smaller files, and `Re-create A Large File` which merges them back together. To access these routines, select `System Utilities` from the main Manager menu, and then select the appropriate function. When breaking up a file, you can either fill up each medium before requesting a new one, or set the size of each file segment.

In particular, the program will require that you do the following:

1. Determine the size of the original file.
2. Decide what size block you want. A size of zero will use the largest space left on the disc. The last block is sized to fit the remainder of the file. Segments need not be identical in size.
3. Select a unique name for each segment.
4. Put the segments back together.

Note

When re-creating a file, put the segments back together in the same order as the original file (since the program has no knowledge of the ordering of parts).

When using these programs remember that each medium must be initialized (with `MEDIAINIT`) and zeroed (with the `Filer`), and each medium should be empty before the `BREAK-UP A Large File` routine is executed.

Both routines will prompt you for all inputs.

9-Track Magnetic Tape and Paper Tape

HP EGS supplies a utility to transfer individual files to and from 9-track magnetic tape and to paper tape. The supported devices are the HP 7974 magnetic tape drive, and the HP 9884A paper tape punch. The magnetic tapes are unlocked and unformatted.

To access this utility, select `System Utilities` from the main Manager menu, and then select `Mag/Paper Tape Utility`. Note that this utility will only do one-file-at-a-time transfer to and from the magnetic tape. It will *not* do media backup. If a local HP-IB disc is used, the tape drive *must* have its own HP-IB interface. If this interface is a Disc Interface, then a DMA card must also be present.

The functions performed by the utility are:

- Transfer a file to paper tape
- Transfer a file to file #n on a magnetic tape
- Transfer a file to the end of a magnetic tape (append to tape)
- Transfer a file from file #n on a magnetic tape
- User-specifiable record size on magnetic tape output
- User-specifiable density on magnetic tape output (device dependent)
- Automatic search of internal HP-IB and disk interface for magnetic tape drive

This utility will prompt you for all inputs.

Character Code Converter

This program converts files using one character code to files using another character code. To access this program select `System Utilities` from the main Manager menu, and then select `Character Code Converter`.

The program is driven by a table that maps one code to another. Each of these table files contain 256 integers, one integer per line, where each integer is in the range 0 through 255. If we are converting file_a from code_1 to code_2, and a 0 in code_1 maps to a 7 in code_2, then the first line of the code_1-to-code_2 conversion table would be a 7.

HP EGS has two conversion tables: EWSYS:asc2ebc.TEXT and EWSYS:ebc2asc.TEXT. These tables convert from ASCII to EBCDIC, and from EBCDIC to ASCII, respectively. Of course, you are free to create your own tables for other code conversions. The following example shows how this conversion works:

Character	ASCII Code	EBCDIC Code
A	65	193
B	66	194
C	67	195

The typical use of this program is to:

- Convert .ASC or .TEXT files to new formats (after conversion these files are no longer human readable).
- Send files to a new system by placing them on magnetic or paper tape or by transmitting them via data-comm links.

Reconfigure System

This program re-configures the menus in the HP EGS manager. Use `Reconfigure System` when you add a new control file from a flexible disc or after you modify a volume search file (suffixed `_vs`) or a control file that defines a menu (suffixed `_cn`).

To use this program, insert the flexible disc containing the new control file into disc drive #3, select `System Utilities` from the main `Manager` menu, and then select `Reconfigure System`.

HP EGS Files and Volumes

The following tables list the files (except drawing files) contained in the volumes of your HP EGS software. These files are necessary to run the HP EGS Graphics Editor, each personality, and each pre- and post-processor. For a detailed list of the drawing files available for each personality, please refer to *Learning HP EGS*.

Operating System Files

SYSVOL:SYSEGS	HP EGS operating system core
SYSVOL:STARTEGS	HP EGS operating system startup file
SYSVOL:HPEGS	HP EGS Manager
SYSVOL:sysmsgs.ms,ASC	HP EGS operating system message file
SYSVOL:INITEGS	HP EGS libraries
SYSVOL:EDITOREGS	HP EGS editor
SYSVOL:FILEREGS	HP EGS filer
SYSVOL:MEDIAEGS	HP EGS media initialization program
SYSVOL:TAPEEGS	HP EGS tape backup program
SYSVOL:TABLEEGS	HP EGS configuration table

Graphics Editor and General Drawing Files

EWSYS:asc2ebc,TEXT	Converts ASCII characters to EBCDIC
EWSYS:ebc2asc,TEXT	Converts EBCDIC characters to ASCII
EWSYS:cconv.ms,ASC	Message file for character conversion program
EWSYS:codeconv,CODE	Character conversion program
EWSYS:diag7974,CODE	Diagnostics program for HP 7974 magnetic tape
EWSYS:diag9884,CODE	Diagnostics program for HP 9884 paper tape
EWSYS:filbm.ms,ASC	Message file for file break and mend programs
EWSYS:filbrk,CODE	File break program to break-up large files
EWSYS:filmnd,CODE	File mend program to re-combine files
EWSYS:swvol,CODE	Utility which lets you stop a stream file and insert a new disc
EWSYS:gdstrt,TEXT	Start file for General Drawing personality
EWSYS:gd_pr,TEXT	Process file for General Drawing personality
EWSYS:gd_mc,TEXT	Macro file for General Drawing personality
EWSYS:gd_mn,TEXT	Screen menu file for General Drawing personality
EWSYS:gds_mn,TEXT	Secondary screen menu file for General Drawing personality
EWSYS:gd_tm,TEXT	Tablet menu file for General Drawing personality
EWSYS:gdtmenu_d	Tablet menu drawing for General Drawing personality
EWSYS:emptmenu_d	Blank tablet menu drawing for General Drawing personality
EWSYS:empt_tm,TEXT	Tablet menu file for blank tablet menu
EWSYS:gd.ms,ASC	Message file for General Drawing personality
EWSYS:gd_mi,ASC	Index to the General Drawing message file
EWSYS:hpeg, CODE	HP EGS Manager
EWSYS:hpeg_cn,ASC	Manager's control file describing the Manager menu
EWSYS:hpeg.ms,ASC	Manager's message file
EWSYS:hpeg_vs,TEXT	Manager's volume search table, which defines where to look for files

20 Files, Volumes, and Mass Storage

<code>EWSYS:hpegs_vsx,TEXT</code>	Alternate volume search table for the configuration (local vs. SRM) you are not using
<code>EWSYS:gedit,CODE</code>	Graphics Editor program
<code>EWSYS:testmfile,CODE</code>	Program to create an index to a message file
<code>EWSYS:gedit_mi,ASC</code>	Index to the Manager's message file
<code>EWSYS:mputil,CODE</code>	Mag and paper tape utility
<code>EWSYS:mputil_ms,ASC</code>	Message file for magnetic and paper tape utility
<code>EWSYS:tpdrv_r_ms,ASC</code>	Message file for tape driver utility
<code>EWCODE:note_st,TEXT</code>	Default note stroke table
<code>EWCODE:text_st,TEXT</code>	Default text stroke table
<code>EWCODE:kata_st,TEXT</code>	Katakana (Japanese) stroke table
<code>EWCODE:roman_st,TEXT</code>	Roman character stroke table, non-proportional spacing
<code>EWCODE:rompp_st,TEXT</code>	Roman character stroke table, proportional spacing
<code>EWCODE:smooth_st,TEXT</code>	Smooth character stroke table, non-proportional spacing
<code>EWCODE:smthpp_st,TEXT</code>	Smooth character stroke table, proportional spacing
<code>EWCODE:note45_st,TEXT</code>	Note stroke table to convert from EGS/45
<code>EWCODE:text45_st,TEXT</code>	Text stroke table to convert from EGS/45
<code>EWCODE:gedit_ms,ASC</code>	Graphics Editor message file, which contains all messages, code-words, peripheral select code information, etc.
<code>EWCODE:add_c,m,CODE</code>	Segmented Graphics Editor commands
<code>EWCODE:arc_c,m,CODE</code>	
<code>EWCODE:are_c,m,CODE</code>	
<code>EWCODE:cal_c,m,CODE</code>	
<code>EWCODE:cat_c,m,CODE</code>	
<code>EWCODE:cdwd_ms,ASC</code>	
<code>EWCODE:cha_c,m,CODE</code>	
<code>EWCODE:codeword,CODE</code>	
<code>EWCODE:copy,c,m,CODE</code>	
<code>EWCODE:cur_c,m,CODE</code>	
<code>EWCODE:dat_c,m,CODE</code>	

EWCODE:dec,m,CODE
EWCODE:delete,CODE
EWCODE:dimdi_c,m,CODE
EWCODE:dimr_c,m,CODE
EWCODE:dims_c,m,CODE
EWCODE:dimu_c,m,CODE
EWCODE:disp_c,m,CODE
EWCODE:dist_c,m,CODE
EWCODE:dum_c,m,CODE
EWCODE:ech_c,m,CODE
EWCODE:edi_c,m,CODE
EWCODE:equ_c,m,CODE
EWCODE:eva_c,m,CODE
EWCODE:exi_c,m,CODE
EWCODE:fil_c,m,CODE
EWCODE:files_c,m,CODE
EWCODE:gather,CODE
EWCODE:gri_c,m,CODE
EWCODE:grp_c,m,CODE
EWCODE:hel_c,m,CODE
EWCODE:ide_c,m,CODE
EWCODE:init,m,CODE
EWCODE:lis_c,m,CODE
EWCODE:loa_c,m,CODE
EWCODE:loc_c,m,CODE
EWCODE:mem_c,m,CODE
EWCODE:modify,CODE
EWCODE:mov_c,m,CODE
EWCODE:nam_c,m,CODE
EWCODE:pac_c,m,CODE
EWCODE:pag_c,m,CODE
EWCODE:pau_c,m,CODE
EWCODE:plot_lib,CODE
EWCODE:poi_c,m,CODE
EWCODE:pre_c,m,CODE
EWCODE:proc_lib,CODE
EWCODE:rem_c,m,CODE
EWCODE:res_c,m,CODE
EWCODE:ret_c,m,CODE
EWCODE:rev_c,m,CODE
EWCODE:sav_c,m,CODE
EWCODE:sea_c,m,CODE

22 Files, Volumes, and Mass Storage

EWCODE:sho_c,m,CODE
EWCODE:size,m,CODE
EWCODE:smash,CODE
EWCODE:stretch,CODE
EWCODE:tme_c,m,CODE
EWCODE:trace_c,m,CODE
EWCODE:uni_c,m,CODE
EWCODE:vol_c,m,CODE
EWCODE:win_c,m,CODE

Electrical Engineering Files

EWEE:ee_cn,ASC	Control file for Electrical Engineering menu
EWEE:ee_cnX,ASC	Alternate control file for configuration (local vs. SRM) you are not using
EWEE:ee_vs,ASC	Volume search table for Electrical Engineering menu
EWEE:ee_vsX,ASC	Alternate volume search table for configuration (local vs. SRM) you are not using
EWEE:pcstrt,TEXT	Start file for PC Board Layout personality
EWEE:pcclstrt,TEXT	Start file for PC Board Layout Connection Lister
EWEE:pc_pr,TEXT	Process file for PC Board personality
EWEE:pc_mn,TEXT	Screen menu file for PC Board personality
EWEE:pcpts_mn,TEXT	Secondary screen menu file for PC Board personality parts
EWEE:pc_tm,TEXT	Tablet menu file for PC Board personality
EWEE:pcmenu_d	Tablet menu drawing for PC Board personality
EWEE:pc_mc,TEXT	Macro file for PC Board personality
EWEE:pc_ms,ASC	Message file for PC Board personality
EWEE:pc_mi,ASC	Index to the PC Board Layout message file
EWEE:schstrt,TEXT	Start file for Electrical Schematic personality
EWEE:schclstrt,TEXT	Start file for Electrical Schematic Connection Lister
EWEE:sch_pr,TEXT	Process file for Electrical Schematic personality
EWEE:sch_mn,TEXT	Screen menu file for Electrical Schematic personality

EWEE:schpts_mn,TEXT	Secondary screen menu file for Electrical Schematic personality parts
EWEE:sch_tm,TEXT	Tablet menu file for Electrical Schematic personality
EWEE:schtmenu_d	Tablet menu drawing for Electrical Schematic personality
EWEE:sch_mc,TEXT	Macro file for Electrical Schematic personality
EWEE:sch_ms,ASC	Message file for Electrical Schematic personality
EWEE:sch_mi,ASC	Index to the Electrical Schematic message file
EWEE:concomp,CODE	Connection list comparison program
EWEE:conmerge,CODE	Connection list merge program
EWEE:cutils_ms,ASC	Message file for connection list merge and compare programs
EWEE:rnest,CODE	Rat's nest generate program
EWEE:rnest_ms,ASC	Message file for rat's nest generate program
EWEE:matlst,CODE	EE material list program
EWEE:matlst_ms,ASC	Message file for EE material list program
EWEE:mlpfe,CODE	Material list parts file editor
EWEE:mlpfe_ms,ASC	Message file for material list parts file editor
EWEE:eePart_Pf	Parts file for EE material list program
EWEE:mlpfconv,CODE	Program to convert old parts files from EGS/45 and EGS/200
EWEE:pfconv_ms,ASC	Message file for EE parts file conversion program

Mechanical Engineering Files

EWME:mestrt,TEXT	Start file for Mechanical Engineering personality
EWME:me_cn,ASC	Control file for Mechanical Engineering personality
EWME:me_cnX,ASC	Alternate control file for configuration (local vs. SRM) you are not using
EWME:me_vs,TEXT	Volume search table for Mechanical Engineering personality
EWME:me_vsX,TEXT	Alternate volume search table for configuration (local vs. SRM) you are not using

24 Files, Volumes, and Mass Storage

EWME:me_pr,TEXT	Process file for Mechanical Engineering personality
EWME:me_mn,TEXT	Screen menu file for Mechanical Engineering personality
EWME:mepts_mn,TEXT	Secondary screen menu file for Mechanical Engineering personality parts
EWME:mesym_mn,TEXT	Secondary screen menu file for Mechanical Engineering personality symbols
EWME:me_tm,TEXT	Tablet menu file for Mechanical Engineering personality
EWME:metmenu_d	Tablet menu drawing for Mechanical Engineering personality
EWME:me_mc,TEXT	Macro file for Mechanical Engineering personality
EWME:me_ms,ASC	Message file for Mechanical Engineering personality
EWME:me_mi,ASC	Index to the Mechanical Engineering message file
EWME:me_mlst,CODE	ME material list program
EWME:mel_ms,ASC	Message file for ME material list program
EWME:mlpfe,CODE	Material list parts file editor
EWME:mlpfe_ms,ASC	Message file for material list parts file editor

Rule Check Post-processor Files

EWRC:cutils_ms,ASC	Message file for connection list merge and compare programs
EWRC:rc_cn,ASC	Control file for rule check post-processors
EWRC:rc_cnX,ASC	Alternate control file for configuration (local vs SRM) you are not using
EWRC:concomp,CODE	Connection list comparison program
EWRC:conmerge,CODE	Connection list merge program
EWRC:rnst_ms,ASC	Message file for rat's nest generate program
EWRC:rnst,CODE	Rat's nest generate program
EWRC:pcbphy,ASC	Default configuration file for Printed Circuit Board connection lister
EWRC:schlog,ASC	Default configuration file for Schematic Logical Port connection lister

EWRC:schPhy,ASC	Default configuration file for Schematic Physical Port connection lister
EWRC:con_ms,ASC	Message file for connection list program
EWRC:verify,CODE	Connection list program
EWRC:con_mi,ASC	Index to connection list message file

Photoplot and Drill Post-processors Files

EWOPT:Phdr,CODE	Photoplot/Drill program
EWOPT:Phdr_cn,ASC	Control file describing Photoplot/Drill menus
EWOPT:Phdr_cnX,ASC	Alternate control file for configuration (local vs. SRM) you are not using
EWOPT:Phdr_def	File containing default answers to program prompts
EWOPT:Phdr_ms,ASC	Message file for Photoplot/Drill program
EWOPT:run_Phdr,TEXT	Example stream file for loading Photoplot/Drill program from the Workstation

IGES Translator Files

EWOPT:toiges,CODE	EGS-to-IGES translator program
EWOPT:toegs,CODE	IGES-to-EGS translator program
EWOPT:iges_cn,ASC	Control file describing IGES translator menu
EWOPT:iges_cnX,ASC	Alternate control file for configuration (local vs. SRM) you are not using
EWOPT:toiges_df,ASC	File containing default answers for EGS-to-IGES translator
EWOPT:toegs_df,ASC	File containing default answers for IGES-to-EGS translator
EWOPT:toiges_ms,ASC	Message file for EGS-to-IGES translator
EWOPT:toegs_ms,ASC	Message file for IGES-to-EGS translator

Using the Text Editor

Chapter 2: Using the Text Editor

When to Use the Editor	27
Entering the Editor	28
Adding Text (Insrt)	29
Confirming or Aborting Commands	31
Refreshing the Screen	31
Moving the Cursor	32
Deleting and Replacing Text	34
Deleting Text (Dlete)	34
Deleting Large Sections of Text (Zap)	35
Typing Over Characters (Xchng)	36
Replacing Character Strings (Rplace)	37
Copying and Moving Text (Cpy)	40
Searching for Text Patterns (Find)	42
Formatting Text	43
Defining the Format (Set environment)	43
Adjusting Margins (Adjst)	45
Filling Paragraphs (Margin)	46
Printing a File	47
Exiting the Editor	48
Finding More Information	49

Using the Text Editor

Chapter

2

This chapter introduces the basics of the text Editor. The Editor enables you to create, change, store, and retrieve both programs and textual documents. The Editor has built-in reminders (prompts) and uses single keystroke commands.

The Editor is a cursor-based screen editor. (The cursor is the blinking underline symbol on the screen.) Cursor movement provides access to any part of a text file so that you can move rapidly through the file to read and edit your text.

The programs and documents created by the Editor are usually stored as TEXT files but can be stored as ASCII files or Data files. Two HP EGS commands, ARCHIVE and GENERATE, save HP EGS drawings as ASCII files. You can use the Editor to change these files.

This chapter demonstrates how to enter and use the Editor by showing you how to type and change a letter. Refer to the Editor Reference section of this manual for a complete description of each Editor command.

When to Use the Editor

Whenever you want to type something—a parts list, a program, an office memo, or a computer manual—think of the Editor.

Entering the Editor

To enter the Editor from the HP EGS Manager, follow these steps:

1. Make sure the HP EGS Manager menu appears on the screen:

```
Use arrow keys to move cursor to desired function.  
Select function with <RETURN> or <ENTER>.  
->System Utilities  
    General Drawing Editor  
    Electrical Engineering  
    Mechanical Engineering
```

2. Move the → to System Utilities and press **Return**. The System Utilities menu looks like this:

```
Use arrow keys to move cursor to desired function.  
Select function with <RETURN> or <ENTER>.  
  
->RETURN To Main Menu  
    Media Initialization  
    CS80 Tape Backup  
  
    Editor  
    Filer  
    Pascal Command Line  
  
    Break-up A Large File  
    Re-create A Large File  
    Mag/Paper Tape Utility  
    Character Code Converter  
    Message File Utility  
  
    Install Codeword  
    Re-boot System  
    Reconfigure System
```

3. Move the → to Editor and press **Return**. The following display appears on the screen:

```
Editor [Rev 3.1  1-Aug-85]  
  
Copyright 1984 Hewlett-Packard Company,  
    All rights reserved.  
  
No workfile found.  
File? (<ret> for new file, <stop> exits)  
:
```

4. This display is the front door of the Editor; it is asking you if there is an existing file that you want to edit. Since you have none, press **Return** to indicate that you want to create a new file. The Editor's command line then appears on the screen:

```
Editor: Adjst Cpy Dlete Find Insrt JMP RPlace XchnG Zap ?
```

You are now inside the Editor.

5. The prompt line shows a partial list of commands available in the Editor. Press **?** to see the rest of the commands:

```
>Edit: Margin Page Set environment Verify ?
```

6. Press **?** again and the main Editor prompt reappears.

Adding Text (Insrt)

Enter the Insrt command by pressing **I**. This command enables you to insert text—that is, type freely—inside the Editor.

The command line for Insrt shows what keys are active, as well as keys for accepting and escaping (aborting) everything typed during the insert session. The command line for Insrt provides the following information:

- **Text** indicates that all typewriter keys are active for inserting text.
- **<bs>**, **<clr ln>** indicates that you may use **Backspace** to simultaneously back up and delete typing errors, and use **Clear line** to erase the line containing the cursor (unless it was the first line inserted).
- **<sel> accepts**, **<sh-sel> escapes** indicates that you press **Select** when you finish typing to make your text permanent, or **Shift** **Select** to erase everything you typed since entering insert mode.
- Use **Return** to make blank lines.

30 Using the Text Editor

To try it out, press and type the following letter as shown (don't worry about correcting errors):

Dear HP:

* Just got my new HP EGS system . * hate typing, but the manual insists that * try out the Editor, so here * am typing this note to you.

So far, * am

Enjoying
Tolerating
Loathing

this manual. * find it

Too easy to follow.
Just about right.
Too hard to follow.

Please accept my heart-felt

Thanks
Disgust

for making my introduction to the HP EGS system

An enjoyable experience.
An endurable experience.
A real nightmare.

Respectfully yours,

Press to exit insert mode. If you had held down while pressing , the entire letter would have been erased.

Confirming or Aborting Commands

The **Select** key tells the Editor to accept all of the insertions or changes you have made in the text file. The cursor remains where it was when you pressed **Select**. Conversely, pressing **Shift** while pressing **Select** (shown as **Shift Select**) tells the Editor to ignore all of the changes made since initiating the command, and leaves the cursor where it was when the command was initiated. Both key sequences (**Select**) and (**Shift Select**) return you to the main Editor prompt.

The additions and changes that you make during an editing session are temporarily stored in your computer during the edit session. You can store them permanently by writing them to a file. At the end of this tutorial you will learn to store your letter in a file.

Refreshing the Screen

Use the Verify, **V**, command to refresh or update the current screen display from memory, move the current line (where the cursor is) to the middle of the screen, and display the Editor prompt. If the cursor is located in the first 23 lines of text when Verify is used, the line containing the cursor is not moved.

Moving the Cursor

The arrow keys, the **Return** key and **Tab** key, the space bar, and the cursor wheel (also called the knob) move the cursor. The wheel normally moves the cursor left or right, depending on which direction you turn it. If you hold down **Shift** while turning the wheel, the cursor moves up or down while remaining in the same column position.

An integer in the range 1 to 9999 can be used as a repeat factor before all of the cursor control keys. (Repeat factors must be in the range 1 to 4095 for use with the **Tab** key.) The result will be the same as if you had pressed the key that many times. For instance, typing the number 42 and then pressing the space bar causes the cursor to move 42 characters in the current direction.

The Jump command offers another means of cursor positioning. Press **J** and the top of your screen displays:

```
>JUMP: Begin End MArker <sh_sel>
```

Typing **B** causes the cursor to jump to the beginning of the file and the Editor's main prompt reappears. Now press **J** then **E**, and the cursor moves to the end of your text.

You can also Jump to previously set markers (see the Set command in the Editor Reference in this manual) by typing **J** **M** followed by a marker name.

The beginning and end of a file are simply the first and last characters in the current text file. The Editor adjusts these internal pointers automatically as the text file is changed.

The Page command enables you to move through a file one screen (23 lines) at a time. If the cursor is not at the end of the file, press **J** **E**. Now type < to change from the forward to the backward direction and press **P** (for Page). The top half of your screen now looks like:

Dear HP:

* just got my new HP EGS system . * hate typing, but the manual insists that * try out the Editor, so here * am typing this note to you.

So far, * am

Enjoying
Tolerating
Loathing

this manual. * find it

Too easy to follow,
Just about right,
Too hard to follow,

Please accept my heart-felt

Thanks
Disgust

for making my introduction to the HP EGS system

Since the cursor movement direction is still backward, press **>** to change it to forward. The Page command is especially handy when moving through a large file.

Deleting and Replacing Text

Three commands are available for deleting and replacing text. Dlete (delete) enables you to remove characters or lines of text, Xchgng (exchange) enables you to type new letters over old ones, and Rplace (replace) enables you to replace a series of characters with a new series of characters.

Deleting Text (Dlete)

The Dlete command is invoked by pressing . It is the opposite of insert, removing every character that the cursor touches.

The command line for Dlete provides the following information:

- > is the direction indicator. It indicates which direction the cursor will move when you press the space bar. When the indicator is pointing to the right (>), the cursor moves forward when the space bar is pressed. When the indicator is pointing to the left (<), the cursor moves backward when the space bar is pressed. To change the direction of the indicator, press and .
- <> indicates that you may use the space bar to move the cursor forward, wiping out characters as it goes.
- <Moving commands> indicates that you may use all of the cursor control keys (the knob, the arrow keys, , and) to move the cursor.
- <sel> deletes, <sh-sel> aborts> indicates that you press when you finish entering text to make your deletions permanent, or to abort your deletions and return the text to its state before the delete operation was begun.

Try this out. Position the cursor somewhere in the list of words:

```
Enjoying
Tolerating
Loathing
```

Choose the word that most closely expresses your attitude and delete the other two. To delete a word, position the cursor under its first letter and press . Then move the cursor forward to erase the word.

Also try moving the cursor back again, and notice that the word reappears. Your deletions are not permanent until you exit delete mode with the key. Now erase the word again, and press . The word is now deleted.

Next, erase the other word in the list that doesn't express your sentiments, but this time, press **Shift** **Select** to exit delete mode. Notice the word is not deleted. Press **D**, erase it again, and press **Select** this time to make the deletion permanent.

Continue deleting all inappropriate words in the other three lists, using the space bar to move the cursor. Experiment with the direction indicator, also. Press **<**, and notice that the cursor moves backward when you hold down the space bar. Press **>**, and the cursor moves forward again.

Next, delete the blank lines that separate your chosen words from the rest of the sentences, but leave the chosen words on separate lines. To do this, position the cursor on a blank line and press **D**. Then press **Return** to move the cursor to the beginning of the following line, and press **Select** to complete the deletion.

If you looked charitably on this manual, your letter should look like this:

Dear HP:

* just got my new HP EGS system. * hate typing, but the manual insists that * try out the Editor, so here * am typing this note to you.

So far, * am
 Enjoying
 this manual. * find it
 Just about right.
 Please accept my heart-felt
 Thanks
 for making my introduction to the HP EGS system
 An enjoyable experience.

Respectfully yours,

Deleting Large Sections of Text (Zap)

Use Zap, **Z**, to delete or move large sections of text. Zap deletes all text between an anchor and the current cursor position and stores it in the copy buffer. The anchor is located at the position in the text where the most recent Adjust, Find, Insert, or Replace command was executed. Confirm the position of the anchor by pressing **=** to move the cursor to the anchor.

Before you use the Zap command, learn to use the Copy the Buffer command (described in the section called "Copying and Moving Text").

Typing Over Characters (Xchg)

The Xchg command is invoked by pressing **X**. It enables you to make corrections by typing new characters over old ones; that is, by exchanging one character for another.

The command line for Xchg provides the following information:

- `Text` indicates that all typewriter keys are active for exchanging text.
- `<bs>` indicates that you may use the **Backspace** key to simultaneously back up and delete typing errors.
- `<sh-sel> aborts, <sel> accepts` indicates that you press **Shift** **Select** to undo everything you have done since beginning the exchange, or **Select** to make your changes permanent.

In the letter, all of your chosen words begin with uppercase letters. Use the Xchg command to change the first letter of each to lowercase.

Position the cursor at the first letter of the first chosen word and press **X**. Now type the lowercase equivalent over the original. Experiment by holding down the **k** key to replace letters on the line with Ks. Notice that you cannot replace letters past the end of the line. Now hold down **Backspace** and watch the original letters return. Also note that you cannot exchange letters previous to the letter where you began the exchange.

Finally, make sure the first letter of the word is lowercase, and press **Select** to make the change permanent. Repeat this procedure for the first letters of all chosen words (use **Return** to move the cursor down through lines of text). When you're through, your letter should look something like this, depending on your choice of words:

Dear HP:

```
* just got my new HP EGS system, * hate typing, but the manual
insists that * try out the Editor, so here * am typing this note to you,
```

```
So far, * am
enjoying
this manual, * find it
just about right,
Please accept my heart-felt
thanks
for making my introduction to the HP EGS system
an enjoyable experience,
```

Respectfully yours,

Replacing Character Strings (Rplace)

The Rplace command is invoked by pressing . It is used to replace occurrences of a string (a series of characters) with a new string.

The command line for Rplace provides the following information:

- L tells the Editor that you are going to give it a literal string. A literal string may be isolated or it may be embedded in a word. The alternative is to tell the Editor you are going to give it a token string which must appear as a word.
- V tells the Editor that you want to operate in verify mode. The verify mode makes changes one at a time after asking you if you want this occurrence replaced.
- <target> (target) tells you to type in the string to be replaced.
- <sub> (substitution) tells you to type in the new string.
- Use a repeat factor to make more than one substitution of the same string. The Editor does not prompt you for the repeat factor; it must be typed before you press to choose Rplace. For example, type 2 /old-string/new string/ to make the replacement two times. Use a slash (/) character to change all occurrences of the string in the current direction. (Remember that cursor movement can be changed with and .)

Use Rplace to replace all occurrences of * in the letter. Type: 6 /*//I/.

The slashes are used to delimit the target and substitution strings. Any non-alphanumeric or non-control characters can be used as delimiters. This is necessary when the slash is part of the target string or substitution string.

After you type the final delimiter, the screen clears and displays:

```
>Rpl[LG]: <sh-sel> aborts, R replaces, ' ' doesn't
```

```
Dear HP:
```

```
*_ just got my new HP EGS system. * hate typing, but the manual
insists that * try out the Editor, so here * am typing this note to you.
```

```
So far, * am
enjoying
this manual. * find it
just about right.
Please accept my heart-felt
thanks
for making my introduction to the HP EGS system
an enjoyable experience.
Respectfully yours,
```

The cursor is positioned behind the first occurrence of the string *. Press **R** and watch what happens.

```
>Rpl[5]: <sh-sel> aborts, R replaces, ' ' doesn't
```

Dear HP:

I just got my new HP EGS system. *_hate typing, but the manual insists that *_try out the Editor, so here *_am typing this note to you.

So far, *_am enjoying this manual. *_find it just about right. Please accept my heart-felt thanks for making my introduction to the HP EGS system an enjoyable experience.

Respectfully yours,

The first string * has been replaced with 'I', the cursor is now positioned behind the second occurrence of the target string, and the prompt shows that you can make five more replacements. Press the space bar (represented by ' ' in the prompt) to leave the string unchanged. The screen displays:

```
>Rpl[6]: <sh-sel> aborts, R replaces, ' ' doesn't
```

Dear HP:

I just got my new HP EGS system. *_hate typing, but the manual insists that *_try out the Editor, so here *_am typing this note to you.

So far, *_am enjoying this manual. *_find it just about right. Please accept my heart-felt thanks for making my introduction to the HP EGS system an enjoyable experience.

Respectfully yours,

The cursor is now positioned behind the third occurrence of * following 'insists that'. Press **R** to replace this one and then press it again to replace the remaining occurrences of *. The screen displays:

```
>Rpl[G]: <sh-sel> aborts, R replaces, ' ' doesn't
```

```
Dear HP:
```

```
I just got my new HP EGS system, * hate typing, but the manual insists
that I try out the Editor, so here I am typing this note to you.
```

```
So far, I am
enjoying
this manual, *_find it
just about right.
Please accept my heart-felt
thanks
for making my introduction to the HP EGS system
an enjoyable experience.
```

```
Respectfully yours,
```

The prompt at the top of the screen tells you that the Editor could not find any more occurrences of the specified string in the file. The cursor is positioned at the final occurrence of the string but it has not yet been changed. Press the space bar. The Editor prompt reappears, the final occurrence of the string gets replaced and, the cursor remains at the same place on the screen.

Position the cursor at the remaining occurrence of * and replace it with I.

Copying and Moving Text (Cpy)

The Cpy command is invoked by pressing **C**. Use this command to copy other files or buffer contents (explained below) into your text. This feature makes it possible to copy and duplicate portions of text.

The command line for the Cpy command provides the following information:

- **Buffer** indicates that you press **B** to copy the contents of the buffer beginning at the position of the cursor.
- **File** indicates that you press **F** to copy the contents of a file beginning at the position of the cursor.
- **<sh-sel>** indicates that you can abort the copy operation by pressing **Shift Select**.

Although you have created no files yet, you can practice to see how this command works. You would first position the cursor where you want the file copied. For instance, if you wanted to copy the file at the end of your letter, you would position the cursor there.

Next, press **C**, then **F** to indicate your desire to Copy a File. The command line changes to:

```
Copy: File[marker,marker] ?
```

The prompt is asking for the complete file specification of the file you want to copy, including the volume ID (or unit number) and the file name. If you had a file to copy, you would type in the file specification and press **Return**. Because you do not, press **Return** to abort the operation.

The copy buffer operation is useful for undoing an inadvertent deletion, and for moving and copying blocks of text. Before you try it out, you must understand the buffer.

The buffer is an area of memory where the text of your last insertion or deletion operation is stored. Every time you perform an insertion or deletion, the old contents of the buffer are replaced with the text that you just inserted or deleted.

To see this, position the cursor at the top of your letter, press **D**, and press **Return** several times to delete several lines of text. Delete the entire letter if you like. Press **Select** to make your deletions permanent. Everything you deleted is now stored in the buffer. Then, before you do anything else, press **C** and then **B** to Copy the Buffer. The deleted lines are restored.

The preceding example shows how to recover text you delete by mistake. Remember that you must discover and correct your mistake before another deletion or insertion is performed.

There is another use for copying the buffer: moving text.

Assume that you decide to make the first paragraph of your letter the last. How do you do this, without retyping the whole paragraph at the end of the letter, and deleting the original paragraph at the beginning? You can use the buffer.

When you delete the paragraph, you effectively copy it into the buffer. Then you can move the cursor to the desired position and copy the buffer back into its new position. Follow these steps:

1. Position the cursor at the end of the letter and insert two blank lines pressing to enter insert mode and pressing twice. Then press to exit insert mode. This separates the paragraph you are going to copy from the rest of the text.
2. Position the cursor at the first letter of the first word in the first paragraph (I). Press , and then press until the paragraph is deleted. Press to accept the deletion.
3. Position the cursor a few lines below the end of the letter. (You can move the cursor around without affecting the buffer; the buffer is only changed when or is pressed.) Press and the entire paragraph is copied.
4. Before you finish, copy the paragraph back into its original position by moving the cursor to the top (just above So) and pressing again. Note that the buffer is unchanged by the last copy operation, allowing you to copy the same paragraph as many times as you wish.
5. Delete the paragraph you copied at the bottom of the letter.

Searching for Text Patterns (Find)

The Find command is invoked by pressing **F**. Use it to find a specific text pattern. This command is similar to the Rplace command in that it looks for the pattern that you specify, beginning at the current cursor location; it also interprets the pattern you specify as a Token or a Literal, according to the current Token environment parameter setting.

The command line for the Find command provides the following information:

- **T** tells the Editor that you are going to give it a Token string which must occur as a word.
- **L** tells the Editor that you are going to give it a Literal string which may occur as a word or part of a word.
- `<target>` tells you to type the string to be found.

Jump to the beginning of the file by pressing **J** and then **B**. Find the first occurrence of the word “my” by pressing **F** and then typing these characters: `/my/`. The slashes (/) act as delimiters for the pattern that you want to find. These delimiters do not have to be the slash (/) character; the Find command uses the first character that you specify as the delimiter (except **T** while in Literal mode and **L** while in Token mode), so you will need to follow the pattern with the same delimiter.

When the pattern is found, the cursor is placed at the end of the pattern. You can use commands to change the text (such as `eXchange`), or you can search for the next occurrence of the pattern by pressing **F** and then **S** (for Find Same).

If the pattern is not found, then this prompt appears:

```
>ERROR: Pattern not found, <space> continues.
```

Press space to answer the prompt, which puts you back into normal Edit mode.

Formatting Text

The Editor allows you to format text with the `Adjst` and `Margin` commands. Text is also formatted by inserting or deleting blanks where needed.

The `Margin` command is active only in the documentation-editing environment. Use the `Set` environment command to define the editing environment.

Defining the Format (Set environment)

The `Set` environment command is invoked by pressing `S`. This command enables you to set several characteristics of the editor, including margins, paragraph indenting, and line filling.

When you press `S`, the following four options appear:

- `Env` tells you to press `E` to set each editing characteristic individually.
- `Mrk` tells you to press `M` to set markers in your text.
- `Prsg` tells you to press `P` to preset the editing characteristics to values suitable to typing in and editing Pascal programs.
- `Doc` tells you to press `D` to preset the editing characteristics to values suitable for typing in and editing letters, memos, manuals, and other documents.

Press `S` `D` (`Set Doc`) to get the following display:

```
>Environment: {options} <sel> or <sp> leaves
  Auto indent   False
  Filling       True
  Left margin   0
  Right margin  78
  Para margin   5
  Command ch    ^
  Token def     False
  Ignore case   False
  Zap markers
  2 bytes used, 728062 available
```

44 Using the Text Editor

This menu shows the suggested editing characteristics for a documentation-editing environment. You can change these characteristics by typing the initial letter of a characteristic, typing the new value, and pressing the space bar to enter the new value. For example, to change the right margin setting from 78 to 60:

1. Press **R** to indicate you want to change the Right margin. Notice that the cursor jumps to this location and the entry goes blank.
2. Type in your new value for the right margin:
60
3. Press the space bar to enter the new value.

Press the space bar to set the environment to documentation-edit mode.

Press **S** **P** (Set Prog) to get the following display:

```
>Environment: {options} <sel> or <sp> leaves
Auto indent   True
Filling       False
Left margin   0
Right margin  78
Para margin   5
Command ch    ^
Token def     True
Ignore case   False
Zap markers
2 bytes used, 728062 available
```

This menu shows the suggested editing characteristics for a program-editing environment. To accept the default settings, press the space bar. The Editor is now set in an environment designed for entering and editing Pascal programs.

Note

For a complete description of all editing environment characteristics, refer to the Editor Reference chapter of this manual.

Adjusting Margins (Adjst)

The Adjst command is invoked by pressing **A**. It provides a means of shifting the starting column of a line of text left or right and helps make text more readable.

The command line for the Adjst command provides the following information:

- **LJust** tells the Editor to move the text to the left.
- **RJust** tells the Editor to move the text to the right.
- **Center** tells the Editor to move the text to the center.
- **<arrow keys>** The arrow keys, the **Backspace** key and the cursor wheel can be used to Adjst text.
- **<sel>** tells you to use **Select** to complete the Adjst command and return to the Editor command line.

Position the cursor at the line that begins with the word “Thanks”. Press **A** then **L**, and use the right cursor arrow key to move the word to the right. Return the word to its original position and press **Select**.

Repeat factors are available for use with the Adjst command so that many lines of text can be shifted at one time.

Note

Avoid using Adjst with large repeat factors because **Shift Select**, which usually aborts all changes made by a command, is not available for exiting the Adjst command. To recover the original format of your text you would have to Adjst it again.

Now that the line is in its original place, press **A** (Adjst), type 3 **▶** (to indent the text three spaces to the right), and then type 6 and press **v**. Watch what happens: the cursor moves down six lines and shifts each line three spaces to the right. Thus, the Adjst command is useful for indenting entire blocks of text.

Press **Select** to terminate the Adjst command. If you wish to make adjustments in other parts of your text file, exit the Adjst command using **Select** before moving the cursor from one area to another. Otherwise you may make unwanted adjustments to your text.

Before you continue, use Adjst to restore the text to the left margin.

Filling Paragraphs (Margin)

The Margin command is invoked by pressing . It is active only in the documentation-editing environment, and is used to clean up a paragraph left messy by using insertions and deletions. The appearance of a paragraph after the Margin command depends on the settings of the editing characteristics.

Before using Margin, you must understand the Editor's definition of a paragraph. A paragraph is any block of text that has a blank line above it and a blank line below it. By this definition the block of text between "so here I am writing this note." and "Respectfully yours," is a paragraph. You can clean it up with the Margin command.

Simply position the cursor anywhere in the paragraph and press . The paragraph is restructured. Notice also that the first paragraph is not indented properly, so position the cursor there and press again.

If Something Goes Wrong...

If you receive the following error message, the editing environment is not set for documents.

```
ERROR: Wrong environment <space> continues
```

Press the space bar, then , then the space bar again, and re-try the Margin command.

When you finish, the letter should look something like this:

Dear HP:

I just got my new HP EGS system. I hate typing, but the manual insists that I try out the Editor, so here I am typing this note to you.

So far, I am enjoying this manual. I find it just about right. Please accept my heart-felt thanks for making my introduction to the HP EGS system an enjoyable experience.

Respectfully yours,

Printing a File

To print a file on a printer attached to your computer, follow these steps:

1. Press **Q** to quit the Editor.

The following prompt appears on the screen:

```
>Quit:
      Update the workfile and leave
      Exit without updating
      Return to the editor without updating
      Write to a file name and return
```

2. Press **W** PRINTER:

or

W #6:

While the file is being printed the message 'Writing' is displayed. When the printer is finished the the following prompt appears on the screen:

```
Exit from or Return to the Editor?
```

3. Press **Return** to re-enter the Editor. (If you press **E** your file is erased from memory.) The following paragraphs explain how to save your file and exit from the Editor.

Exiting the Editor

Use the following procedure to exit the Editor:

1. Make sure the Editor's command line appears on the screen:

```
Editor: Adjst Cpy Dlete Find Insert Jmp Rplace Xchng Zap ?
```

If it does not, press **Return**, **Select**, or **ESC**.

2. Press **Q** to quit the Editor. The following prompt appears on the screen:

```
>Quit:
      Update the workfile and leave
      Exit without updating
      Return to the editor without updating
      Write to a file name and return
```

This list shows all the ways you can exit the editor. Each option is described below:

- **Update the workfile and leave.** This option will save the text you created in the workfile. The workfile is a file the operating system creates for you on the system volume. It is used as a temporary file for storing programs or documents while they are being developed. Workfiles are discussed in more detail in the next section.
 - **Exit without updating.** This option exits the editor without saving the text you created there. It should be used only if you want to discard the work in your last editing session.
 - **Return to the Editor without updating.** This option simply returns you to the Editor, at the same point in your text where you exited it.
 - **Write to a file name and return.** This option allows you to save your work in a file of your choosing.
3. For this example, choose the last option by pressing **W**. The following prompt appears:

```
Name of output file (<ent> to return) -->
```

Type:

```
MYVOL:LETTER Return
```

The text is saved in the file named LETTER.TEXT. Notice the string of dots moving across the screen, indicating that the file is being written to the disc.

The Editor automatically appended the suffix `.TEXT` to your file, so its full name on the disc is `LETTER.TEXT`. This suffix tells you that the file has been stored using the text format, which includes not only your words, but also the editing environment in force when the file was saved. Since you exited the Editor with the environment set in Doc mode, these settings will automatically be restored when you return `LETTER.TEXT` to the Editor for editing.

You could have also saved this file using the ASCII file format, by specifying the file name as `LETTER.ASC` rather than as `LETTER`. The ASCII format saves your file as a string of ASCII characters; no environment information is saved.

4. After the file is written to the disc, the size of the file is displayed (in bytes), and you are asked:

Exit from or Return to the editor ?

5. Press to exit to the System Utilities Menu. (returns you to the editor.)

You are now back at the System Utilities Menu.

Finding More Information...

To learn more about the Editor, refer to the Editor Reference chapter of this Manual.

Using the Filer

Chapter 3: Using The Filer

Entering the Filer	52
Manipulating Volumes and Directories.....	53
Listing Volumes (Vols).....	53
Changing the Default Volume (Prefix-vol).....	54
Listing Directories (Ldir)	55
More Directory Information (Ext-dir)	58
Creating Files	59
Manipulating Files.....	59
Copying and Moving Files (Filecopy).....	59
Changing File Names (Change)	60
Removing Files from a Volume (Remove).....	61
Changing File Types and Printing Files (Translate).....	61
Introducing Workfiles.....	63
Copying a File into a Workfile (Get).....	63
Copying a Workfile Back to the Original File (Save)	65
Clearing a Workfile (New).....	66
Displaying the Workfile Name and State (What)	66
Managing Local Mass Storage.....	67
Checking Mass Storage Readability (Bad secs).....	67
Compacting Files in a Volume (Krunch)	67
Creating an Empty Directory (Zero).....	68
Using the SRM File Commands	69
Using Passwords to Protect Files (Access).....	69
Making Directories and Files (Make)	69
Making New Pointers to a File (Duplicate link)	69
Changing the Working Directory (Udir).....	69
Exiting the Filer.....	70
Finding More Information.....	70

Using The Filer

Chapter

3

This chapter introduces the basics of the Filer. The Filer contains tools that enable you to manipulate files and volumes. You used the graphics personalities and text editor to create the data stored in files. Use the Filer to manipulate packages of data (files) and groups of files (volumes).

The commands described in this chapter are grouped according to function. The first group of commands enables you to list volumes and directories and to specify the default volume. These commands include: Volumes, List Directory, Extended Directory, and Prefix.

File manipulation commands comprise the second group. These commands enable you to copy and remove files, change file names, and translate files from one type to another. These commands include: Filecopy, Remove, Change, and Translate.

The third group of commands allows you to manipulate workfiles, which are used like system default files. These commands include: Get, New, Save, and What.

The fourth group of commands are maintenance commands to be used on local mass storage. These commands include: Bad-secs, Krunch, and Zero.

The next group of commands can be used only if your system is connected to a Shared Resource Management (SRM) system. These commands include: Access, Duplicate link, Make, and Udir.

The last command, Quit, explains how to exit the Filer subsystem.

File system concepts are explained in Chapter 1: *Files, Volumes, and Mass Storage*; each Filer command is described in depth in the Chapter 6: *Filer Commands*.

Entering the Filer

To enter the Filer from the HP EGS Manager, follow these steps:

1. Make sure the HP EGS Manager menu appears on the screen:

```
Use arrow keys to move cursor to desired function,
Select function with <RETURN> or <ENTER>,
->System Utilities
    General Drawing Editor
    Electrical Engineering
    Mechanical Engineering
```

2. Move the → to System Utilities and press **Return**. The System Utilities menu looks like this:

```
Use arrow keys to move cursor to desired function,
Select function with <RETURN> or <ENTER>.
```

```
->RETURN To Main Menu
    Media Initialization
    CS80 Tape Backup

    Editor
    Filer
    Pascal Command Line

    Break-up A Large File
    Re-create A Large File
    Mag/Paper Tape Utility
    Character Code Converter
    Message File Utility

    Install Codeword
    Re-boot System
    Reconfigure System
```

3. Move the → to Filer and press **Return**. The following command line appears on the screen:

```
Filer: Change Get Ldir New Quit Remove Save Translate Vols What Access Udir ?
```

You are now inside the Filer.

4. The command line shows only a partial list of the available commands; to see the others, type **?**. The Filer's alternate command line appears below:

```
Filer: Bad-secs Ext-dir Krunch Make Prefix-vol Filecopy Duplicate Zero ?
```

Type **?** again and the main Filer prompt reappears.

Manipulating Volumes and Directories

This group of commands provides information about the file system and enables you to change the default volume. The examples which follow are for an HP EGS system using local mass storage. If you have an SRM system your files will not be organized the same way. Ask your SRM system manager where the HP EGS files are stored.

Listing Volumes (Vols)

The Volumes command is invoked by pressing **V**. It is used to list all volumes that are on-line.

Press **V**. The display will look similar to the following, depending on your particular system configuration:

```
Volumes on-line:
 1  CONSOLE:
 2  SYSTEM:
 6  PRINTER:
11 * SYSVOL:
12 # EWSYS:
13 # EWCODE:
14 # EWEE:
15 # EWME:
16 # EWRC:
17 # EWOPT:
18 # MYVOL:
19 # V19:
Prefix is - MYVOL:
```

The left column lists all unit numbers; the right column lists each unit number's associated volume ID. The # between the columns indicates a blocked device; that is, a device capable of storing files, such as a disc drive or a bubble memory card. The volume marked with an asterisk (*) is the system volume; the prefix (default) volume appears at the bottom. The default volume is the volume ID the computer will assume if you do not specify a volume ID when naming a file.

Certain unit numbers are permanently assigned. These include:

- Unit #1, volume ID CONSOLE:. This is your computer screen.
- Unit #2, volume ID SYSTEM:. This is your computer keyboard.
- Unit #6, volume ID PRINTER:. This is your printer.

Note that all of these devices are unblocked; they cannot store files. Saying that they have a volume ID may seem odd, because we described volumes earlier as a collection of files. HP EGS uses this convention for the sake of consistency, so that all devices may be referenced similarly (by unit number or volume ID). Think of unblocked volume IDs as identifiers that allow you to access the devices.

Changing the Default Volume (Prefix-vol)

The Prefix-vol command is invoked by pressing . It is used to change the prefix volume (default volume).

The prefix volume is another name for the default volume. The Filer calls it the prefix volume; the What and Version commands in the Main Command Level call it the default volume. When you perform an operation on a file in the prefix volume the volume name is provided by the system so that you do not have to type it in. For example, if SYSVOL: is the prefix volume, you could edit MYVOL:LETTER only by typing in the full file specification (volumename:filename). If you use the Prefix-vol command to make MYVOL: the prefix volume, you would have to type only LETTER to edit MYVOL:LETTER.

To prefix to a different volume (that is, reset the default volume), use the Prefix-vol command while inside the Filer.

To use Prefix-vol, press to get the following prompt:

```
Prefix to what directory ?
```

Type:

```
MYVOL: 
```

The computer responds with:

```
Prefix is MYVOL:
```

To prefix to a unit number rather than to the volume residing in the unit, follow this same procedure. (For example, you could have typed #18: to change the prefix to MYVOL:.)

Listing Directories (Ldir)

The Ldir command is invoked by pressing **L**. It is used to obtain a list of all files in a specified directory.

Press **L** to get the following prompt:

```
List what directory?
```

Type:

```
SYSVOL: Return
```

The directory of the SYSVOL: volume is then displayed, and should be similar to this:

```
SYSVOL:                Directory type=LIF level 1
created 4-APR-85        15,07,33 block size=256
changed 8-APR-85        08,12,13 Storage Order
...file name...        #blks   #bytes   last chng
SYSEGS                 54      13824
TABLEEGS               52      13312
HPEGS                  562     143872
STARTEGS               184     47104
syseds.ms.ASC         6        1536
INITEGS                880     225280
FILER                  226     57856
EDITOR                 234     59904
MEDIINIT.CODE         140     35840
TAPEBKUP.CODE         70      17920
```

```
Files shown=10 allocated=10 unallocated=70
```

```
Blocks (256 bytes) used=2408 unused=1765 largest space=1765
```

The following paragraphs explain the information available in a directory listing:

Line 1:

```
SYSVOL:                Directory type= LIF level 1
```

This line tells you the name of the volume whose directory this is, namely SYSVOL:. It also tells you the directory type.

Line 2:

```
created 4-Apr-85      15,07,33 block size=256
```

This line tells you the date and time the volume was created, in this case, April 4, 1985 at 33 seconds past 3:07 in the afternoon (your SYSVOL: volume may have a different inception date). It also tells you that a block of disc storage is equal to 256 bytes. A block is merely a unit of disc storage space; file sizes are expressed in the number of blocks they occupy on the disc.

Line 3:

```
changed 8-Apr-85      08,12,13 Storage order
```

This line tells you the date and time the volume was last changed. It also tells you that the files are listed in the order in which they are stored on the disc. The alternative is alphabetic order, discussed later in this section.

Line 4:

```
..file name...      # blks      # bytes      last chng
```

This line is the heading for the list of files that follows. The `file name` is the name of the file. The `# blks` is the number of blocks of disc storage occupied by the file, a measure of its size. The `# bytes` is the number of bytes of disc storage occupied by the file. Because a block is 256 bytes, this number is the previous column multiplied by 256. The `last chng` column tells when the file was last changed; that is, read from and written back to the disc.

Note

The dates displayed in the directory are read from the system time and system date that you set after booting the HP EGS system. Therefore, be sure to keep the time and date current.

Line 5

```
SYSEGS              54      13824  3-Apr-84
```

SYSEGS is a file on the SYSVOL: volume. By default, files are listed in the order in which they were stored; the file you used last would be listed first.

Line 16

```
FILES shown=10 allocated=10 unallocated=70
```

This line tells you that ten files are `shown` in this listing; that there are ten entries `allocated` in the directory, and that 70 entries remain `unallocated`.

Here, the listing is showing you how many of those directory entries have been filled by files, and how many are left before the directory (and hence the volume) is full. Understand that you can run out of directory entries but still have room left on your disc. Volume storage management will be discussed more in the Krunch command section of this chapter.

Line 17

```
BLOCKS (256 bytes) used=2408 unused=1765 largest space=1765
```

The preceding line told you how much directory space was allocated on the volume; this line describes how much file storage space remains.

The `used` field tells you how many 256-byte blocks of disc storage have been used by the files listed. `unused` tells you how many blocks remain before the disc runs out of file storage space. The `largest space` gives the largest contiguous hole in the unused storage space. If you try to write a file to the disc that is larger than the largest space, the file cannot fit in this hole and thus cannot be written. The Krunch command can be used to widen this hole, however, as described later.

One final note before we move on: you can list the files in alphabetic order by typing `[*]` after the unit number or volume ID. To list the SYSVOL: volume in alphabetic order, press

L and then type:

```
SYSVOL:[*] Return
```

The alphabetical listing produced should resemble this:

```
SYSVOL:                Directory type=LIF level 1
created 4-Apr-85       15,07,33 block size=256
changed 8-Apr-85       08,12,13 Storage Order
...file name...       #blks   #bytes   last chng
EDITOR                 234     59904
FILER                  226     57856
HPEGS                  562    143872
INITEGS                880    225280
MEDIAINIT.CODE        190     35840
STARTEGS              184     47104
SYSEGS                 54     13824
TABLEEGS               52     13312
TAPEBKUP.CODE         70     17920
sysegs_ms.ASC         6       1536
```

```
Files shown=10 allocated=10 unallocated=70
```

```
Blocks (256 bytes) used=2408
```

Note

If a listing is too long to fit on the screen, the listing will pause after displaying the first few files. Press the space bar to see the remaining files.

More Directory Information (Ext-dir)

The Extended Directory command provides more information about the files in a volume than a list directory provides. This command would most likely be used by a system manager to identify files that are not being used and to troubleshoot files that cannot be used.

The column entries for each file include: file name, number of blocks used for storage, the file size in bytes, the number of the block where the file starts, the date the file was changed, the file type as recognized by the file system, the type-code used by the directory system, SRM access information the date the file was created, and two extension fields.

Refer to Chapter 6 for more information on this command.

Creating Files

Use the text and graphics editors to create files. Commands such as Save, Overwrite, Archive, and Generate store textual and graphical information in files.

The Make command can be used to reserve space for a file but it does not initialize or change the contents of the space. Make is most useful for creating SRM directories.

Manipulating Files

This group of commands enables you to copy and move files, change file names, remove files from a volume, and change file types.

Copying and Moving Files (Filecopy)

The Filecopy command is invoked by pressing **F**. It is used to make copies of files or entire volumes. In this example, we will demonstrate how to copy a single file.

You will copy the file LETTER.TEXT in MYVOL: to the DRAW: volume.

1. Press **F** to invoke the Filecopy command. The computer displays the following prompt:

```
Filecopy what file ?
```

2. Type:

```
MYVOL:LETTER.TEXT Return
```

The computer prompts:

```
Filecopy to what ?
```

3. Type:

```
DRAW:$ Return
```

The \$ is a wildcard. It tells the computer to use the same file name for the new copy of the file. It is equivalent to typing DRAW:LETTER.TEXT, but is more convenient.

The computer copies the file from volume MYVOL to volume DRAW, and displays the following message when completed:

```
MYVOL:LETTER.TEXT ==> DRAW:LETTER.TEXT
```

Changing File Names (Change)

The Change command is invoked by pressing . It is used to change the name of a file.

To see how it works, change the name of LETTER.TEXT in volume DRAW: to JUNK.TEXT:

1. Press to get the following prompt:

Change what file?

2. Type in the location and name of the file you want to change:

DRAW:LETTER,TEXT

The computer prompts:

Change to what?

3. Type:

JUNK,TEXT

Note

Type in only the new file name; do not type in the unit number again. You are only changing the name of an existing file, not moving it to another volume. Thus, the location of renamed file is the same as the location of the original file.

The computer confirms the change with:

DRAW:LETTER,TEXT ==> DRAW:JUNK,TEXT

4. You may further confirm the change by listing the directory of the DRAW: volume, if you wish. Just press , then type:

DRAW:

Removing Files from a Volume (Remove)

The Remove command is invoked by pressing **R**. It is used to remove a file from a volume.

Use Remove to delete the file JUNK.TEXT from the DRAW: volume. Press **R** to get the following prompt:

```
Remove what file?
```

Type the location and name of the file to be removed:

```
DRAW:JUNK,TEXT Return
```

The computer responds with:

```
DRAW:JUNK,TEXT removed
```

Confirm that JUNK.TEXT is no longer on the DRAW: volume by listing the directory.

Changing File Types and Printing Files (Translate)

The Translate command is invoked by pressing **T**. It is used to translate one file type into another and to print files on the printer.

A file type is determined by the file suffix. The suffix of JUNK.TEXT tells you that JUNK is a text file. The suffix is part of the file name, so JUNK.TEXT and JUNK.ASC are different files. The suffix is appended when the file is created. The file type, once determined, is stored in the directory along with the file name. That means the file type would not be changed if you later changed the file name and removed or changed the suffix. You can see the file type of each file by listing the directory using the Filer's Extended Directory List command.

The file system recognizes the following suffixes:

- .ASC Information stored in a .ASC (ASCII) file is stored as individual strings. Each string has a two-byte string length header.
- .TEXT Text files are produced by the Editor. They have a 1024-byte header containing environment information. This header is followed by the text.
- .CODE Information is stored in a .CODE file as object code which may be executed as a program.
- .SYSTEM This is a special file recognized by the Boot ROM as a file containing an operating system.
- .BAD A .BAD file is used to cover failed disc sectors.

62 Using the Filer

A file whose name at the time of creation does not end in one of these suffixes is said to be of type DATA.

Use Translate to print LETTER.TEXT. Press to get the following prompt:

```
Translate what file ?
```

Type:

```
MYVOL:LETTER,TEXT 
```

The computer prompts:

```
Translate to what ?
```

Recall from the Volumes command that the unit number of the printer is #6:. Type:

```
#6: 
```

The computer translates LETTER.TEXT into a format suitable for the printer.

Note

If your printer failed to print LETTER.TEXT, make sure it is turned on and properly connected to the computer.

Introducing Workfiles

It is important to understand how workfiles are used in this system.

On the surface, workfiles are like any other file. The workfile, a text file named WORK.TEXT, is stored on the system volume. What makes the workfiles different is the way the subsystems use them.

In the Editor, the initial display looked like this:

```
EDITOR [REV 3.0 4-Jun-84]
Copyright 1984 Hewlett Packard Company
  All rights reserved.
No Workfile found.
File? (<ret> for newfile, <stop> exits)
:
```

Notice that the fourth line of this display says:

```
No workfile found.
```

The Editor looked first for a workfile (a file named WORK.TEXT on the system volume). Because it did not find it, it asks you for the name of the file you want to edit. If it had found the workfile, it would have automatically loaded it. The Editor assumes you want to edit the workfile, if it exists.

Copying a File into a Workfile (Get)

The Get command is invoked by pressing . It is used to associate a file name with the workfile. Getting a file, is like copying it into the workfile, where it becomes the default file for all Pascal subsystems. Try this exercise:

1. Press and type MYVOL: to prefix to MYVOL:.
2. Press to invoke the Get command. The computer responds with:

```
Get what file?
```

3. Type the following response to assign the file LETTER.TEXT to the workfile WORK.TEXT:

```
LETTER
```

4. The computer finds the file LETTER.TEXT and reports:

```
Source file loaded
```

5. This statement may be misleading. It implies that the source file, LETTER.TEXT, has been copied into the workfile, WORK.TEXT. You can see this is not true by listing the directory of the system volume:

- a. Press to invoke the List command. When the computer asks:

```
List what directory?
```

type the following to list the system volume (remember, the special symbol * represents the system volume):

```
* 
```

- b. The computer displays the directory listing of the system volume SYSVOL:. Your listing should look similar to this:

```
SYSVOL:                Directory type=LIF level 1
created 4-APR-85       15,07,33 block size=256
changed 8-APR-85      08,12,13 Storage Order
..file name...      #blks   #bytes   last chng
SYSEGS                54      13824
TABLEEGS              52      13312
HPEGS                 562     143872
STARTEGS              184     47104
sysseg$ms,ASC         6        1536
INITEGS               880     225280
FILER                 226     57856
EDITOR                234     59904
MEDIAINIT.CODE        140     35840
TAPEBKUP.CODE         70      17920
```

```
Files shown=10 allocated=10 unallocated=70
```

```
Blocks (256 bytes) used=2408 unused=1765 largest space=1765
```

WORK.TEXT, the workfile, is not there.

6. The file LETTER.TEXT is not copied into WORK.TEXT until the workfile is updated – one of the exit options of the Editor. Press to quit the Filer, and press to enter the Editor.
7. Notice that the Editor did not prompt you for the name of a file to edit; it simply read in LETTER.TEXT, which has been identified as the workfile. Press to quit the Editor, and then to select “Update the workfile and leave” as your exit option.

8. Now press to re-enter the Filer, and list the system volume again as you did in step 5. The directory now contains the updated workfile, WORK.TEXT:

```

SYSVOL:                Directory type=LIF level 1
created 4-Apr-85        15,07,33 block size=256
changed 8-Apr-85       08,12,13 Storage Order
...file name...       #blks   #bytes   last chng
SYSEGS                54      13824
TABLEEGS              52      13312
HPEGS                 562     143872
STARTEGS             184      47104
syssegs_ms,ASC        6        1536
INITEGS              880     225280
FILER                 226     57856
EDITOR                234     59904
MEDIAINIT.CODE       140     35840
TAPEBKUP.CODE        70      17920
WORK.TEXT              8        2048
Files shown=11 allocated=11 unallocated=69
Blocks (256 bytes) used=2416 unused=1757 largest space=1757

```

Copying a Workfile Back to the Original File (Save)

The Save command is invoked by pressing . It is used to save the workfile WORK.TEXT into a permanent file.

Save is the opposite of Get. If you use Get to copy a file into WORK.TEXT, you can use Save to copy WORK.TEXT back into its original file. Try it with your current workfile:

1. Press to invoke the Save command. The computer prompts:

```
Save as LETTER ? (Y/N)
```

2. The computer remembers that your current workfile was retrieved from the file LETTER.TEXT in MYVOL:, so it's asking if you want to save it back in the same file. Respond by typing .

The computer prompts:

```
MYVOL:LETTER,TEXT exists ... Remove/Overwrite/Neither ? (R/O/N)
```

Because the file LETTER.TEXT already exists in the volume in unit MYVOL, the computer wants to know what you want to do with the old file. Your options are:

- Press to first remove the old file, then write the new file.
 - Press to write the new file directly over the old one.
 - Press to abandon the Save operation altogether.
3. Press to remove the old file and write the new one.

Clearing a Workfile (New)

The New command is invoked by pressing . It is used to clear the workfiles, WORK.TEXT.

If there was no way of clearing the workfiles, the Editor would always assume you wanted to edit WORK.TEXT, and would read it in automatically. The only way around this is to clear the workfile, and force the Editor to ask you for the name of the file you wish to edit.

To clear the workfiles, press to invoke New. If you have not used Save to permanently store the workfile, the following message may appear:

```
Throw away current workfile ? (Y/N)
```

Answer , and the workfile is cleared.

To confirm that the workfile is gone, list the directory of the DRAW: volume. Notice that WORK.TEXT has been removed.

Displaying the Workfile Name and State (What)

The What command displays the name and state (saved or not saved) of the workfile. The command is initiated by pressing and requires no other input. The command shows the name of the current workfile or indicates that it is not associated with a file name. It also shows whether the workfile has been saved since the last update to the file.

Managing Local Mass Storage

The commands in this group should be used only if your system is running on a local mass storage device.

Checking Mass Storage Readability (Bad secs)

The Bad sector command enables you to check a mass storage medium to find out if each block (sector) is readable. Refer to Chapter 6 for additional information.

Compacting Files in a Volume (Krunch)

Do not use this command if your system is on a Shared Resource Management (SRM) system.

The Krunch command is invoked by pressing . It is used to compact all files in a volume into one contiguous block, putting all unused storage space in one contiguous hole at the end.

After storing, deleting, and updating files on a volume several times, the storage space becomes fragmented: chopped into small, discontinuous segments. Eventually, you cannot write another file to the volume because there is not a single hole available that is large enough to contain it. The computer will display the message:

```
No room on volume
```

The solution is to Krunch all of the files to the front of the volume, joining all of the free space into one large hole at the back.

To demonstrate the Krunch command, press . The following prompt appears:

```
Crunch what directory ?
```

Type in the name of the volume to be compacted; in this case:

```
MYVOL: 
```

The computer asks again:

```
Crunch directory MYVOL ? (Y/N)
```

Press . The operation begins, and the computer displays:

```
Crunch of directory MYVOL in progress  
DO NOT DISTURB!!
```

There is a reason for this emphatic message. If you open the disc drive door, remove the disc, or lose power during this operation, at least some of your files could be lost.

When the crunch is completed, the computer reports:

```
Crunch completed.
```

If you are ever refused when trying to write a file to a volume, write the file temporarily to a different volume, crunch the volume that refused you, then Filecopy the file from the other volume.

Creating an Empty Directory (Zero)

The Zero command creates an empty directory on the specified volume. The Zero command is not allowed on SRM volumes. (See the Make command.)

Refer to Chapter 6 for more information on the Zero command.

Using the SRM File Commands

Most of the Filer commands already described can be used whether your HP EGS system is operating on a local disc or a Shared Resource Management (SRM). The commands listed in this section are useful only if your system is on an SRM. (Make file can be used on either type of system but Make directory can be used only on SRM systems.)

Using Passwords to Protect Files (Access)

Use Access to protect your files. Passwords can deny or limit access to the file. For example, if you are using HP EGS to design printed circuit boards, you might limit access to your parts library so most users could copy parts but not change or add them.

Making Directories and Files (Make)

The Make command creates files and directories. The Make command reserves space for files; in general, use graphics and text editors to create files. Use the Make command to create SRM directories. For example:

```
Make File or Directory? (F/D)
```

Answer the question by typing and specify the directory location and name. The directory path tells where you want the directory located and its name is the name on the end of the path. For example, if you had an SRM directory:

```
#5:/USERS/MYVOL
```

and you wanted to create a directory to store files created during a graphics project, you would type:

```
#5:/USERS/MYVOL/DRAW
```

The DRAW directory is created in the MYVOL directory.

Making New Pointers to a File (Duplicate link)

The Duplicate link command establishes a new pointer to a file. Refer to Chapter 6 for more information.

Changing the Working Directory (Udir)

Do not use this command; use the Prefix-volume command instead. The Unit directory command changes the volume name and working directory for an SRM unit. The Prefix command performs the same operation but sets the default volume to the SRM volume.

Exiting the Filer

Use the following procedure to exit the Filer:

1. Make sure the Filer's command line appears on the screen:

```
Filer: Change Get Ldir New Quit Remove Save Translate Vols What Access Udir ?
```

2. Press to quit the Filer.

You are now back at the System Utilities Menu.

Finding More Information...

For details on these and other Filer commands, refer to the Filer Reference chapter of this manual and your SRM documentation.

Pascal Command Line Reference

Chapter 4: Pascal Command Line

Information Provided	71
The Pascal Command Line.	72
Pascal Command Quick Reference	73
Hpegs	74
Initialize	75
Memvol	76
Newsysvol.	78
Permanent.	79
Run	80
Stream.	81
User restart	84
Version	85
What	87
eXecute	89

Pascal Command Line

Chapter

4

Information Provided

The next three chapters comprise a reference describing the commands available from the Pascal command line. These commands are part of the operating system supplied with your HP EGS software. HP EGS contains a subset of the Pascal operating system, which does not include the Assembler, Compiler, Debugger, and Librarian. These four programs are not needed to run HP EGS.

This reference contains a quick reference for the commands and an alphabetized description of the command syntax and options. For information on interpreting the syntax diagrams, see the introduction to the *HP EGS Syntax Reference*. Remember that Pascal commands are unlike the other commands presented in this syntax reference, since they are not part of the HP EGS Graphics Editor. Pascal commands allow you to manipulate text, ASCII, and data files from outside the Graphics Editor.

This reference is organized in three sections:

- **Pascal Command Line** (Chapter 4) – The Pascal command line provides access to the commands in the Pascal operating system. The command line enables you to perform simple operations and to access subsystems (like the Editor and Filer).
- **Editor** (Chapter 5) – The Editor is a subsystem of the Pascal operating system that allows you to create, change, store, and retrieve both programs and textual documents.
- **Filer** (Chapter 6) – The Filer is a subsystem of the Pascal operating system that allows you to manipulate files in various ways including moving, listing, duplicating, creating, and deleting files.

The Pascal Command Line

The Pascal command line consists of two lines shown on your screen, only one of which is displayed at a time. You can press the key to switch between them. The uppercase letters in the command line indicate which key to press to start the operation. All of the operations on both lines are available, regardless of which line is being displayed.

The HP EGS operating system contains all the operations available with the Pascal operating system, except the Assembler, Compiler, Debugger, and Librarian. If you have used the Pascal operating system, you may notice that the words in the HP EGS command line are in a slightly different order than those provided with the full Pascal operating system. In the HP EGS Pascal command line, the words appearing on the first line refer to those operations used most often in HP EGS:

```
Heeds Editor Filer eXecute Init Permanent Stream Version What
```

The words appearing on the second line refer to operations that are seldom used or that work only if you have installed the full Pascal operating system in addition to HP EGS:

```
Assembler Compiler Debugger Librarian Memvol Newsysvol Run User
```

If you try to use a command that does not work in your operating system, HP EGS will display an error message telling you that the operation you have selected is not available.

Following is a quick summary of what the various commands in the Pascal command line do:

Pascal Command Quick Reference

Command	Description
Assembler	Calls the Assembler to translate an Assembler language source program into object code. Valid only if the Pascal operating system is installed.
Compiler	Calls the Compiler to compile Pascal source code into object code. Valid only if the Pascal operating system is installed.
Debugger	Runs a program under control of the Debugger. Valid only if the Pascal operating system is installed.
Editor	Calls the Editor for creating or editing a source program or textual document.
Filer	Calls the Filer for management of the File System.
Hpegs	Returns you to the HP EGS Manager menu.
Initialize	Initializes the File System (but not discs).
Librarian	Calls the Librarian for managing, linking, or unassembling object-code files. Valid only if the Pascal operating system is installed.
Memvol	Sets up a memory resident mass storage volume for fast access.
Newsysvol	Asks for a volume to be designated as the system volume.
Permanent	Asks for and permanently loads a code file into memory.
Run	Runs the workfile (compiling it if needed) or the last program compiled since power-up. If there is no workfile, Run operates like eXecute.
Stream	Asks for and executes a stream file whose characters are interpreted as keyboard input.
User restart	Restarts the last program or subsystem that the user ran.
Version	Allows setting the time and date, and displays current system version information.
What	Displays the system file table and allows you to change the system files or system and default volumes.
eXecute	Asks for and runs a code file.

Hpegs

The Hpegs command returns you to the main Manager menu inside HP EGS.



When you choose this command, the system first opens the Manager's control file (EWSYS:hpegs_cn.ASC) to read the menus. It then re-opens the control files inside the volumes listed in the Manager's volume search table. This is the same action that happens when you first enter HP EGS.

For a complete description of control files and volume search tables, refer to *Customizing Your System* in the *Understanding HP EGS* manual.

Initialize

The Initialize command updates Unit Table entries for all units that are currently on-line. The Unit Table contains a record for each of 50 possible logical units available to the File System.



Note

This command *does not* initialize mass storage media; that function is performed by using the MEDIAINIT.CODE utility program. See Chapter 1: *Files, Volumes, and Mass Storage* for further details on initializing media.

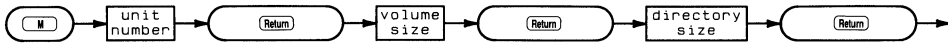
When a device is added to your system after the computer has been powered-up, the Initialize command may in some cases be sufficient to get the system to recognize the new device.

Initialize also performs a device clear for all on-line devices and causes the system to forget the last loaded file (meaning the User command can't reload the last program). The Initialize command also causes all temporary files to be removed from each volume the next time a file is opened on the volume.

The volumes CONSOLE: (Unit #2) and PRINTER: (Unit #6) are special cases; the system always assumes these volumes are on-line.

Memvol

The Memory volume command creates a mass storage volume in memory.



Item	Description/Default	Range Restrictions
unit number	integer	7 through 50
volume size	integer indicating the number of 512-byte blocks	≥ 1
directory size	integer indicating the maximum number of files in the volume	≥ 1

The Memvol command gives you the capability for very fast mass storage operations. This may be useful if you are using a user-written application in conjunction with HP EGS.

When the Memvol command is given, you are first prompted for a unit number. This number corresponds to an entry in the Unit Table. Don't give a unit number which is already in use. The Volumes command in the Filer subsystem shows which unit numbers are currently used. For most applications 50 is the recommended unit number to use for your first memory volume.

You are then prompted for the number of (512-byte) blocks needed for the memory volume. Try to estimate conservatively the amount of memory you want reserved for the memory volume because it cannot be returned for general purpose use without turning off the computer. On the other hand, if you do not specify enough space, you have to create another larger volume.

If you are transferring a source file from disc (as opposed to starting from scratch) you can determine its size by getting a directory listing of the volume that contains it. However, note that different directories return the file size in different units.

- LIF directories use 256-byte "blocks"
- SRM directories use 1-byte "blocks"

LIF is the default directory access method (DAM) for memory volumes.

You are then prompted to give the number of directory entries you need for this memory volume.

```
Number of directory entries ?
```

Type the number you think you will need and press **Return** .

You can refer to your memory volume by its unit number (for example, #50:). Alternately, you can refer to it by its given volume name, which is initially RAM: .

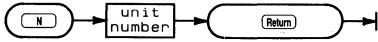
If you plan to use more than one memory volume, use the Change command in the Filer to give each memory volume a unique name.

Note

Because memory volumes are volatile, remember to save the files in the memory volume on a disc before turning off the computer.

Newsysvol

The Newsysvol command specifies a new system volume and updates the operating system file table accordingly.



Item	Description/Default	Range Restrictions
unit number	integer	1 through 50

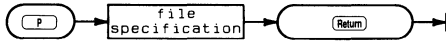
The system file table is used in locating operating system files. It contains the volume and file names of system files (such as EDITOR and FILER). When you press a key at the Pascal command line that invokes one of these subsystems (such as), the system attempts to load the corresponding file indicated in the system file table (the EDITOR file).

You can use this command to specify a new system volume. You are first prompted for a unit number. The device corresponding to the specified unit number is considered to be the new system volume, and serves as a starting point in the search for the system files: ASSEMBLER, COMPILER, EDITOR, FILER, LIBRARIAN, LIBRARY, and the workfile. If any of these system files is not found, the Unit Table is used in a sequential search for the rest of them. As each file is found, the name of the volume on which it is found is appended to the file name (for instance, SYSVOL:LIBRARY), and the complete file specification is placed in the file table. If any system file is not found in this search, the operating system assumes that it will find the file on the flexible disc volume on which it was delivered (for instance, ACCESS:EDITOR).

Use the What command in the Pascal command line to see the resultant system file table.

Permanent

The Permanent command loads a program permanently into memory.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	any legal file specification

The Permanent command can be used to load a user program, a system program (such as Editor or Compiler), or a module that is needed by a program. This code file is then ready to execute immediately when the command is given. A “P-loaded” (Permanently loaded) program does not have to be loaded from disc each time it is run.

After you give the Permanent command, you are prompted for the name of the file that contains the module or program. You do not need to include the `.CODE` suffix; if you do not include one, the suffix will be appended to the file name. If the file to be P-loaded does not have a `.CODE` suffix, end the file specification with a period to suppress the suffix from being appended automatically to the file name.

Several programs may be P-loaded into memory. The operating system keeps track of which programs have been P-loaded. When you give a command to run a program, the operating system checks whether it has been P-loaded; if so, it is executed immediately. If not, it is loaded from disc and then executed; after execution, the memory used by the program is reclaimed.

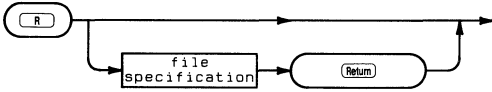
Note

The volume name is not retained when a file is P-loaded. Attempting to execute a file of the same name but on a different volume will still result in the P-loaded file being executed.

For SRM users, do not use a directory path name to execute a P-loaded file.

Run

The Run command executes the workfile or last compiled program.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	any legal file specification

When the Run command is given, the operating system checks whether there is a workfile. If there is a CODE workfile, it is executed; if not, the most recently compiled or assembled file is executed. If there is a TEXT workfile but no CODE workfile, the TEXT workfile is first compiled (with the system compiler) to a CODE file and then the CODE file is executed. If there is no workfile or previously compiled program, the command operates like the eXecute command and you are prompted for a file specification.

Stream

The Stream command executes a file of ASCII characters as if they were being typed from the keyboard.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	any legal file specification

A command stream or stream file is a file that is interpreted as input to the Pascal command line and/or its subsystems in place of keyboard input. The Stream command causes a file to be interpreted. Therefore, a stream file is useful for executing a sequence of commonly used commands without requiring any operator intervention.

A stream file is created with the Editor and may be of type TEXT, ASC, or Data. If you do not specify a suffix, .TEXT is automatically appended to the file name; if the name of the file to be streamed does not have a suffix, add a trailing period to the file name to suppress the suffix.

To generate a valid sequence of keystrokes, you should first run through the desired sequence while noting the keystrokes entered. Note particularly the occurrences or absences of the **Return** key. Then enter the same keystrokes in your stream file. If, during an Editor or Filer command sequence, you encounter an unpredictable question that has a (Y/N) or (R/O/N) question associated with it, do not answer the question in the stream file. These kinds of questions are answered automatically as the file is streamed. (Y/N) questions (Yes/No) are answered Y. (R/O/N) questions (Remove/Overwrite/Neither) are automatically answered R.

After all the characters in a stream file have been interpreted, control returns to the keyboard.

Comments in Stream Files

Stream files may contain comments. A line beginning with an asterisk (*) will be interpreted as a comment if it occurs at the Pascal command line. (Comments cannot be embedded among commands for subsystems or user programs.) When the command interpreter encounters one or more comment lines while streaming, they are displayed briefly on the screen, thus allowing the process to be monitored.

Immediate Execute Keys

If it is necessary to use keys that also act as immediate-execute commands in the Editor, such as **Select** or **Backspace**, use the following key sequences to generate those key-strokes.

Immediate-Execute Key	Generate with These Keys
Select	CTRL Select C
Backspace	CTRL Select H
Tab	CTRL Select I
Clear display	CTRL Select L
Left arrow	CTRL Select H
Right arrow	CTRL Select <
Up arrow	CTRL Select ?
Down arrow	CTRL Select J

If you have a different keyboard, substitute **EXECUTE** or **EXEC** for **Select** in the preceding table.

Prompts for Keyboard Input

A stream file can be made to display a prompt on the CRT and then wait for an input string from the keyboard. The input string is assigned to a variable in the stream file. When the variable is encountered during streaming, the string is used in its place.

This input prompting must appear in the stream file before all of the commands or comments. Up to 36 prompts are allowed. They are denoted with = as the first character on a line.

To prompt for an input string, place an equal sign, followed by a single alphanumeric character variable name (uppercase and lowercase letters used for variables are treated as equal), followed by the prompt text. For example:

```
=f What is the name of the file to be P-loaded ?
```

When the file name is typed in response to the prompt, it is stored in the specified variable, in this case the variable named f.

After the input prompting, begin entering the commands in the stream file. When you want the input string to be given to the operating system, use the variable preceded by @. For example, the following characters are a command stream:

```
P@f
```

The command is the Permanent load command with a file name parameter indicating which file is to be P-loaded. The file whose name was given in response to the above prompt is then P-loaded.

Disabling the Prompt Feature

If the stream file name contains a [*] specifier, the ability to prompt for keyboard input is disabled.

(Normally when a file is Streamed, the file is copied to the file named STREAM on the current system volume; during this copy, prompts are displayed and @ variables are assigned values input from the keyboard by the computer operator. After all variables have been assigned, the file is read as keystrokes; in other words, *STREAM is the file that is actually streamed. The [*] suppresses the normal processing of the prompts and input variables, as the keys are read directly from the specified file.)

User restart

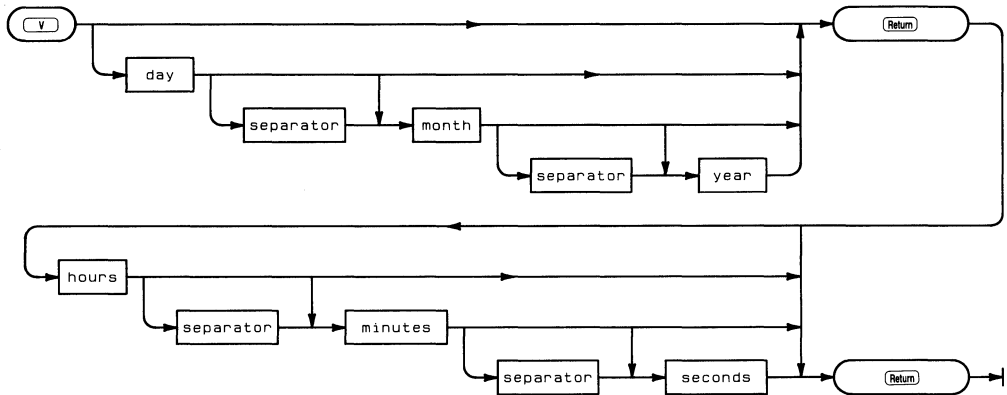
The User restart command causes the last program you ran to be re-run.



The use of “program” includes user programs and subsystems such as Editor, Filer, and Compiler.

Version

The Version command allows you to change the system time and date.



Item	Description/Default	Range Restrictions
day	integer	1 through 31
month	three alpha characters; letter case is ignored	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
year	integer	0 through 99
hours	integer	0 through 23
minutes	integer	0 through 59
seconds	integer	0 through 59
separator	non-alphanumeric character	: - / space, etc.

In addition to the prompt for the system time and date, some operating system information is displayed: the current operating system revision, available global and user memory space information, default and system volume information.

New system date ?

System date is 1-Jul-85
Clock time is 14:14:50

Workstation Rev. 3.0 15-Apr-84

Available Global Space 22818 bytes
Total Available Memory 1168720 bytes

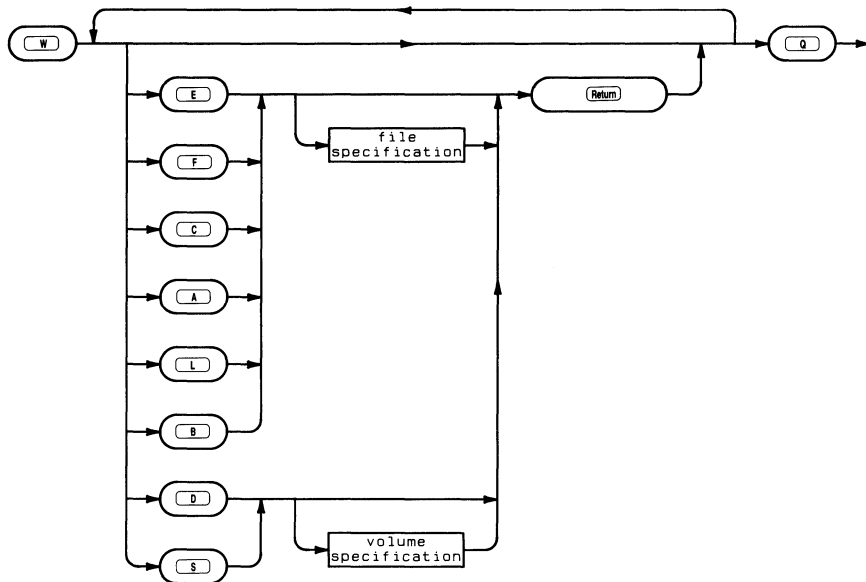
System volume: BOOT:
Default volume: EWSYS:

Copyright 1985 Hewlett-Packard Company.
All rights are reserved. Copying or other
reproduction of this program except for archival
purposes is prohibited without the prior
written consent of Hewlett-Packard Company.

The Version Prompt

What

The What command displays the system file table and allows you to specify new file specifications for the system files.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	any legal file specification
volume specification	complete volume name	any legal volume specification

The system file table contains file specifications used by the operating system when locating system files (Assembler, Compiler, Editor, Filer, Librarian, Library, and Default and System volumes).

```

Assembler Compiler Editor Filer Librarian
liBrary System volume Default volume Quit

```

```

ASSEMBLER   ASM:ASSEMBLER
COMPILER    CMP:COMPILER
EDITOR      BOOT:EDITOR
FILER       BOOT:FILER
LIBRARIAN   LIB:LIBRARIAN
LIBRARY     SYSVOL:LIBRARY

```

```

* System volume:  BOOT:
: Default volume: EWSYS:

```

The What Display

Typing one of the uppercase letters at the top of the menu allows you to change the corresponding file specification for that system file.

Note

When specifying a system file name that does not have a .CODE suffix, use a period at the end of the file name to prevent a .CODE suffix from being appended to the file name.

With this command, you can specify a file other than EDITOR (like HP TechWriter) as the system's text editor. If you change the file name here, then you need only press to invoke the new text editor.

Specifying a logical unit number, such as #3:, as the Default volume allows any disc media in a unit with removable media to be the desired volume. To subsequently specify any volume in the default unit, only the file name need be specified. To do this, make sure that the drive door is open, type for a Default volume change, and then type the following:

```
#3: 
```

eXecute

The eXecute command runs a specified code file.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	any legal file specification

When executing a code file, the file you specify should be compiled or assembled and ready to run (this required previously using the Pascal operating system's Assembler or Compiler). You do not need to include the .CODE suffix in the file name, because the suffix is automatically appended to the file name if not included. If the actual file name does not contain a .CODE suffix, you will need to terminate the file specification with a period to suppress this suffix.

Editor Reference

Chapter 5: Editor Reference

Information Provided	91
Editor Command Quick Reference	92
Adjust	93
Copy	95
Delete	97
Find	99
Insert	102
Jump	105
Margin	106
Page	107
Quit	108
Replace	111
Set	115
Verify	120
eXchange	121
Zap	123

Editor Reference

Chapter

5

Information Provided

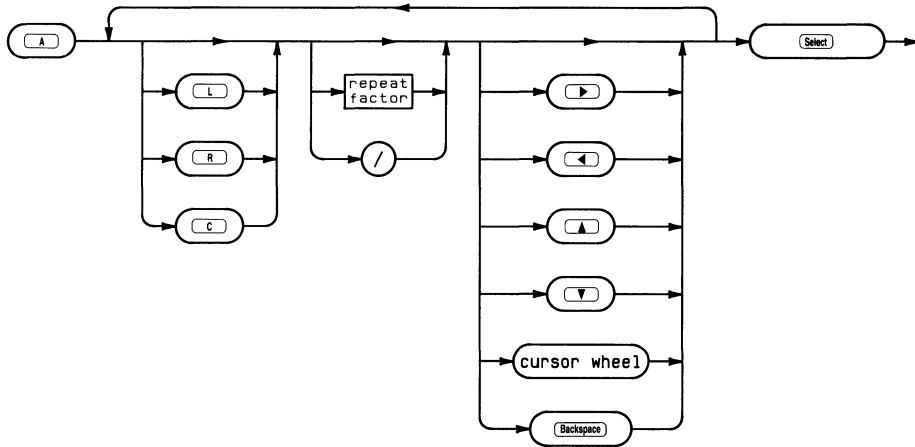
This section describes the Editor commands available from the HP EGS operating system. It contains a quick reference for the commands and an alphabetized description of the command syntax and options. For information on interpreting the syntax diagrams, see the introduction to the *HP EGS Syntax Reference*.

Editor Command Quick Reference

Command	Description
Adjust	Adjusts the column in which a line (or lines) start.
Copy	Inserts text from the copy buffer or from an external file in front of the current cursor location.
Delete	Removes text from the current cursor location to the location of the cursor when Select is pressed.
Find	Positions the cursor after the specified target string.
Insert	Inserts text in front of the current cursor location.
Jump	Positions the cursor at the beginning, end, or specified marker.
Margin	Formats the paragraph where the cursor is located to the margins in the current environment.
Page	Positions the cursor ± 23 lines from the current location.
Quit	Leaves the Editor in an orderly manner. Provides various options for saving the text currently in memory.
Replace	Replaces the specified target string with the specified substitute string.
Set	Modifies the environment or sets markers in the text.
Verify	Updates the displayed text to reflect the text stored in memory.
eXchange	Replaces the text at the cursor with text typed from the keyboard, on a character-by-character basis.
Zap	Deletes all text between the anchor and the current cursor location. (The anchor is set at the location of the latest Adjust, Find, Insert, or Replace command.)

Adjust





The Adjust command horizontally shifts the starting column of one or more lines of text.














Item	Description/Default	Range Restrictions
repeat factor	integer	1 through 9999

The Adjust prompt is:

```
>Adjust: Ljust Rjust Center <arrow keys> [<sel> to leave]
```

The Adjust command provides a means of formatting text and enables you to make text more readable. Adjust uses the line position of the cursor when the command is entered as a starting point. Adjust is a line-oriented command that lets you shift an entire line of text to the left or right using the , , **Backspace**, or cursor wheel. Repeat factors can be used with these keys to shift the text. For example, pressing   results in the line of text shifting 7 spaces to the right.


Pressing  (for Adjust) and then , , or  moves the line to the left or right margin, or centers the line between the two margins. The margins used by these options are the Right and Left margins currently set in the environment (see Set command).

Typing a repeat factor and  or  causes that number of lines to be adjusted the same amount as the accumulated adjustments at that point. The slash (/) functions as an infinite repeat factor and can be used with  and . It causes adjustments to be made from the current line to either the beginning or the end of the text file, respectively. For example, pressing    causes all the text between the current cursor position and the end of the file to be Centered according to the current margins.

Note

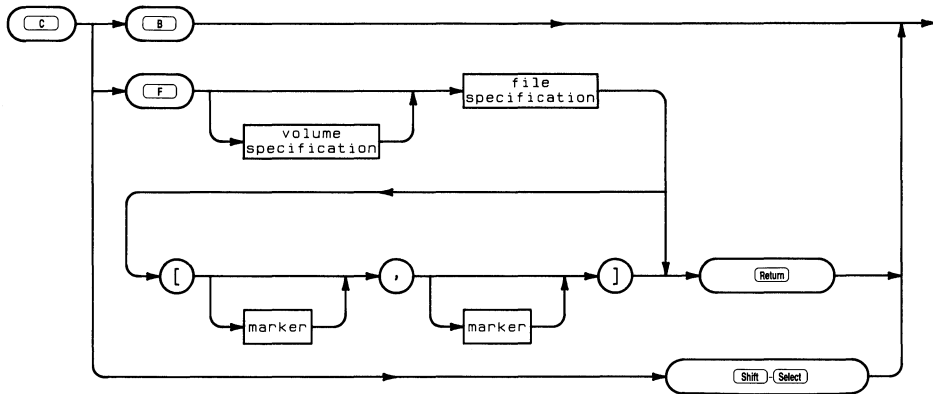
Take care when using large repeat factors or the slash (/) character to adjust text, because the effects of the Adjust command cannot be aborted. Whatever adjustments are made become permanent unless the Adjust command is used again.

Adjust also sets the anchor used by the Zap command. Pressing = moves the cursor to the position of the last Adjust unless the anchor has been reset by either a Find, Insert, or Replace command.

Leave the Adjust command by pressing . The system stores the adjusted text in the computer's memory, and the Editor prompt reappears.

Copy

The Copy command inserts text from a specified file or from the copy buffer.



Item	Description/Default	Range Restrictions
volume specification	complete volume specifier	any valid volume name
file specification	complete file specifier	any valid file name (do not enter .TEXT suffix)
marker	characters that denote a specific location in a file	1 to 8 ASCII characters, excluding CHR(0) through CHR(31), and CHR(127)

The Copy prompt is:

```
>Copy: Buffer File <sh-sel>
```

The Copy command provides a way of moving or duplicating text in a file and copying text from another file. Copy has the Buffer and File options. Buffer refers to the most recent text inserted or deleted; file refers to any file name you specify.

Pressing **C** (for Copy) and **B** (for Buffer) results in the contents of the copy buffer being inserted into your text at the current cursor position. The screen displays the new text and the Editor prompt.

The copy buffer is filled with the text involved in the most recent Delete, Insert, or Zap command. Its contents are cleared with the Margin command. Margin clears the copy buffer regardless of the settings in the environment. Doing a Copy (from a File) also clears the copy buffer. A subsequent Copy Buffer command generates the message:

```
>ERROR: Invalid copy, <space> continues,
```

Any subsequent Delete, Insert, or Zap refills the buffer (destroying its previous contents), and copying from a file clears the contents of the buffer. However, doing a Copy (from Buffer) does not alter the buffer's contents. Neither do any cursor control movements or commands. Therefore, you can make multiple copies of the same text in different locations by repeatedly positioning the cursor and pressing **C** **B**.

To Copy from a file, press **C** **F**. The screen displays:

```
>COPY: File [marker,marker] ?
```

The Editor is requesting a file specification and two marker names. The volume name may be omitted if the file in question is on the default volume. The volume (specified or default) must be on-line. Specification of the two previously set markers (see Set command) is optional but, if given, the marker names must be enclosed in square brackets and separated by a comma. Remember, only .TEXT type files have markers.

If markers are specified, only the text between those two markers is copied. If no markers are specified, the entire file is copied. Only one marker has to be specified. If it is the first marker (that is, followed by a comma), the text is copied from the marker position to the end of the specified file. If only the second marker is given (that is, preceded by a comma), the text is copied from the beginning of the specified file to the position of the marker. The copy occurs at the cursor's position when the Copy command was entered. You can exit the command before all specifications are complete by pressing **Shift** **Select**.

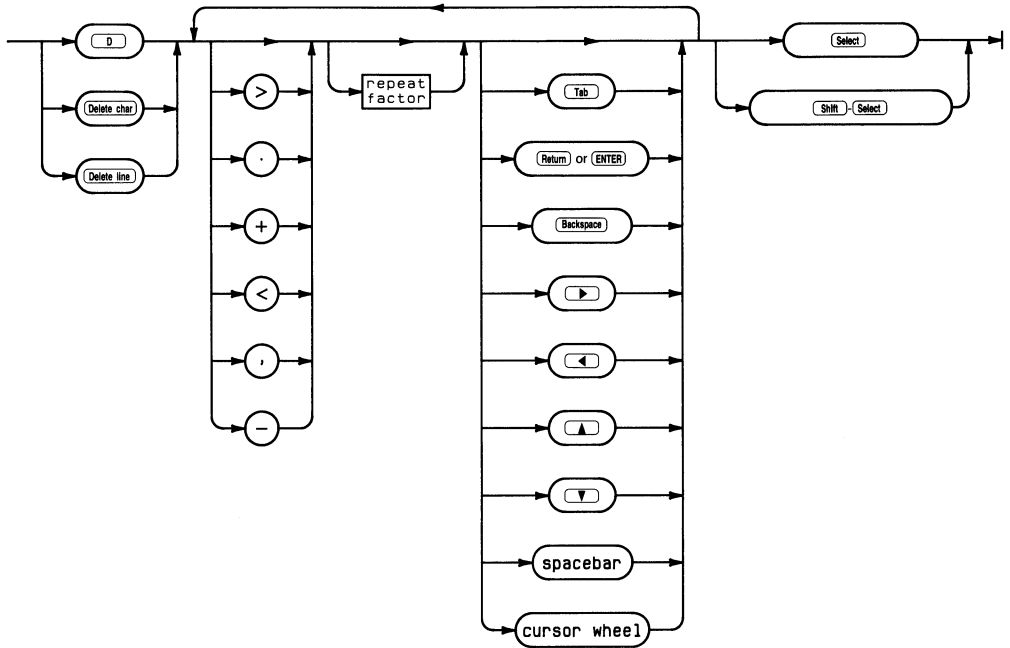
After you type the appropriate information and press **Return**, the Editor displays:

```
>COPY...
```

This shows that the specified text is being copied into your current text. When the operation is complete, the Editor prompt reappears and the screen displays all or part of the text that was copied.

Delete

The Delete command removes text from the current file.



Item	Description/Default	Range Restrictions
repeat factor	integer	1 through 9999 (1 through 4095 for Tab)

The Delete prompt is:

```
>Delete: < > <Moving commands> [<sel> deletes, <sh-sel> aborts]
```

The Delete command enables you to remove text and fills the copy buffer with the deleted text. Delete uses the cursor position when the command is entered as a starting point. Subsequent cursor movement by any cursor control key causes text to be removed between this point and the new cursor position. Text can be recovered by moving the cursor back toward the starting point.

Direction applies in the Delete command and is shown by > (forward) or < (backward) in the Delete prompt. Movement generated with **Tab**, **Return**, and the space bar takes place in the direction shown; forward movement is from the cursor toward the end of the file, and backwards movement is from the cursor toward the beginning of the file. Direction can be changed while in the Delete command by pressing > , or + (for forward) and < , or - (for backward).

Repeat factors are available within the Delete command. For example, pressing **D** (for Delete) and then 9 **Return** will remove 9 lines of text in the current direction, starting at the cursor position.

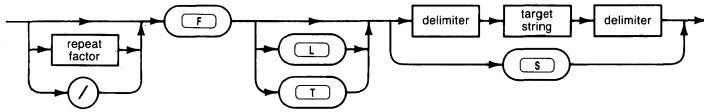
Delete fills the copy buffer with the deleted text and thus provides a means of moving or duplicating text.

To exit the Delete command press **Select** or **Shift Select**. **Select** confirms the deletion, returns the Editor prompt and displays the cursor at its position when **Select** was pressed. **Shift Select** aborts all changes made since Delete was entered, returns the Editor prompt and displays the cursor at its position when Delete was entered.

Note that the copy buffer is filled by whatever is deleted. Whether the command is exited with an **Select** or **Shift Select** makes no difference to the copy buffer.

Find

The Find command moves the cursor to an occurrence of a specified string.



Item	Description/Default	Range Restrictions
repeat factor	integer	1 through 9999
delimiter	character separator that signifies the beginning or end of a string	any non-alphanumeric character; must be used in matched pairs
target string	characters you want to find	1 through 128 characters

The Find prompt is:

```
>Find[l]: L <target>=>
```

or

```
>Find[t]: T <target>=>
```

The prompt displayed depends on whether the “Token” definition in the Editor’s environment is set to True or False. If set to True, the first prompt is displayed; if False, the second is shown. These are explained further later in this section.

In its simplest form, the Find command is executed by pressing **F** and specifying a string surrounded by delimiters. Upon typing the final delimiter, the cursor is positioned at the end of the first occurrence of the specified string in the current direction, if found. If the pattern is not found, then this message is displayed: `ERROR: Pattern not found, <space> continues.` Pressing the space bar returns you to Edit mode at the previous location in the file.

The Find command moves the cursor and sets the anchor (which can be used by Zap) at the location of the target string. In this context, a “string” is a contiguous series of non-control ASCII characters surrounded by delimiters. Delimiters are separators that signify to the Editor the beginning and end of the string. They can be any non-alphanumeric characters such as / ' , and \.

Don't use a delimiter that appears in your string. Delimiters must be matched pairs; if you use (#) to signify the beginning of a string, you must use (#) to signify the end of the string. The maximum length of a target string is 128 characters.

The Find prompt shows the current direction. When searching for a string occurrence, Find looks for that string between the cursor position when the command was entered and either the end of the file (if direction is forward >) or the beginning of the file (if direction is backward <).

Repeat factors are available with the Find command but must be typed before the Find command is initiated. If a repeat factor is used, the Editor positions the cursor at the end of that occurrence of the string. For example, typing

/the/

results in the cursor being positioned at the end of the eighth occurrence of the. The slash / operates in a similar way but signifies the last occurrence of the specified string in the current direction. If no repeat factor or slash character is specified, the repeat factor defaults to the value 1 and the Editor attempts to find the first string occurrence. The Find prompt displays this value in square brackets.

After pressing , the prompt on your screen contains either an L or T for “literal” or “token” modes. Literal and token are mutually exclusive; if one option is shown as available, the other is automatically the default. If L is shown in the prompt and you want to use the token search mode, simply type in the target string surrounded by delimiters. The search will take place in the default mode (in this case, token). To do the same search in the literal mode, press then type in the string as before. The Find command then searches for a literal form of the string.

A literal string is exactly that – a literal string of characters either isolated or embedded in a word or paragraph. A token string is one which is isolated by delimiters. Delimiters in this context are any ASCII characters except numbers or alphabetic characters. Blanks, commas, and periods are common delimiters in English text because they separate words.

To illustrate literal and token searches, the following example assumes the direction is forward (>) with the cursor located at or before the start of the sentence shown. In the sentence

That's my hat!

a token search for hat moves the cursor behind the last word hat in the sentence, whereas a literal search for hat moves the cursor behind the hat embedded in That's .

The “same” option is another feature of the Find command. Same refers to the most recent target string used in either the Find or Replace command. Suppose you type the sequence **F** **L** *galactic*. After pressing the final delimiter (*), the Editor moves the cursor behind the first literal occurrence of the target string galactic. Then typing **F** **L** **S** results in the cursor moving behind the next literal occurrence of the same target.

Note

If a Replace has been done since the last Find operation, the target string used by the “same” option is now the target specified in the Replace command.

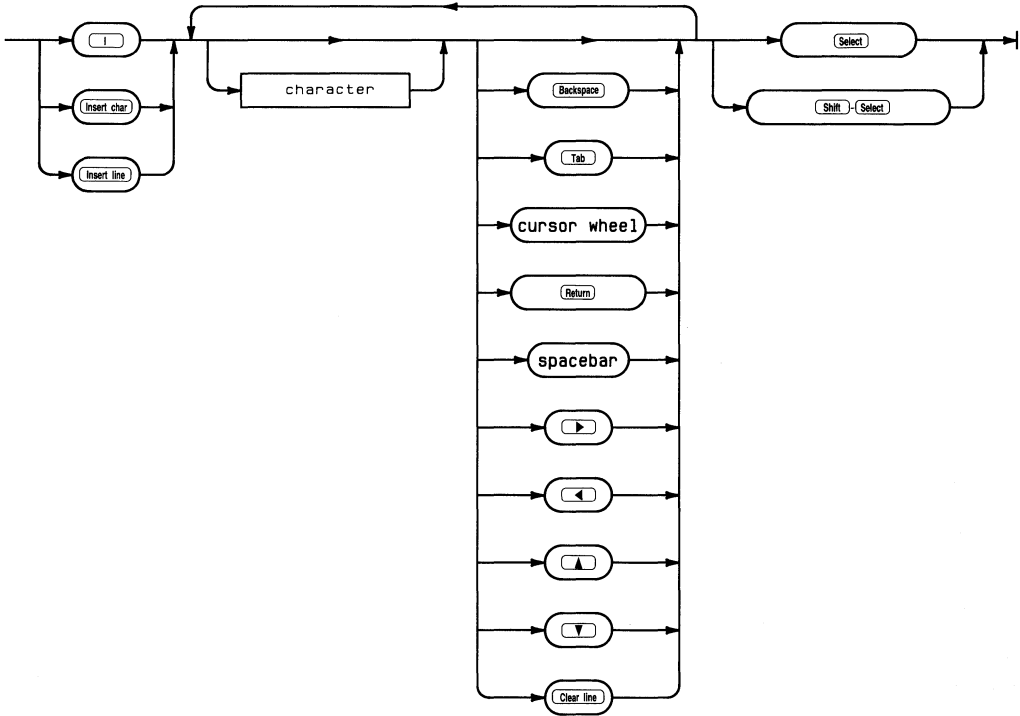
Searches are sensitive to the case of the characters (uppercase and lowercase) unless Ignore case and Token are set to True in the Environment. Type **S** and **E** to Set the Environment. Type **I** and **T** to set Ignore case to True. Type **T** and **T** if Token is not already True. After these two conditions are met, the Editor treats both the target string and each token string as uppercase.

Find is one of the commands that sets the anchor used in the Zap command (meaning it moves the cursor to the beginning of the most recent item adjusted, found, inserted, or replaced).

Shift **Select** can be used to abort the Find command before all specifications are complete. If **Shift** **Select** is used, the target pattern used by the “same” option remains unchanged. **Select** cannot be used with the Find command. The command is executed immediately when the final delimiter (or **S** if “same” is used) is typed.

Insert

Insert opens a window in the current file for the subsequent insertion of text.



Item	Description/Default	Range Restrictions
character	any printable character	recommended values are ASCII characters CHR(32) through CHR(126)

The Insert prompt is:

```
>Insert: Text <bs>, <clr ln> [<sel> accepts, <sh-sel> escapes]
```

The Insert command opens a window in the text file directly in front of the cursor position for text creation. When initiated by pressing **I**, Insert char, or **Insert line**, the text from the cursor to the right edge of the screen is shifted to the right. Insertion always takes place directly in front of the cursor location when Insert was entered. Any sequence of non-control ASCII characters may be inserted and any cursor control key may be used. However, unless the movement generated by the cursor control keys is backward, question marks (?) appear in the text. You can **Backspace** to delete a character or press **Clear line** to delete the most recently inserted line. **Clear line** is available only after a line of text has been inserted. Backspacing past the point at which Insert was entered is not possible.

The way in which text insertion takes place depends on flags or parameters set in the Editor's environment. These flags have default values supplied by the Editor but can be changed with the Set command. The ones of concern here are Filling and Auto indent. These two options generally have opposite values. Most of what you need to know about Filling and Auto indent can be summed up in one sentence: If you are writing program source text, set Filling to false and Auto indent to True (Program mode); if writing regular text, set Filling to True and Auto indent to False (Document mode).

Filling, when set True, performs both "wrap around" and "margin" functions. As inserted text approaches the Right margin (another environment option), the Editor attempts to fit the words on the current line. If a word would cause the line to extend beyond the right margin, it is automatically shifted to the next line (that is, the system supplies a carriage-return and a line-feed). When the insertion is completed by pressing **Select**, all text following the cursor in the current paragraph is margined. Margining adjusts the text to fit between the environment's margins and also compresses blanks in the text. You can have two blanks following these four characters: ? , : ! . All other blanks are compressed into a single blank character.

Note that the Editor's definition of a paragraph is *any* text delimited by *any* combination of blank lines, lines having the Command character as the first non-blank character in a line, or the beginning and end of a text file. The Command character is yet another of the environment's options; see the Set command for more details.

Note

As the definition of a paragraph infers, the Editor does not distinguish tables from other kinds of text material. Any insertions within a table will result in the table being margined (collapsed) if Filling is set to True and the insertion is exited with **Select**. Use the Set command to set Filling to False before inserting in a table or list. (**Shift** **Select** will NOT restore the text to its original state.)

If Filling is False, a beep is generated as you approach the end of the line, signaling you to press **Return** the same way a bell on a typewriter does. If you continue to insert text past the last visible column on your screen, the Editor accepts the characters and shows you that they are outside of the display area by placing an exclamation point (!) in the final column. To access these characters, complete the insertion by pressing **Select**, position the cursor before the last visible word on the line, and press **I** followed by **Return** to insert a carriage-return (and thus break the line at that point).

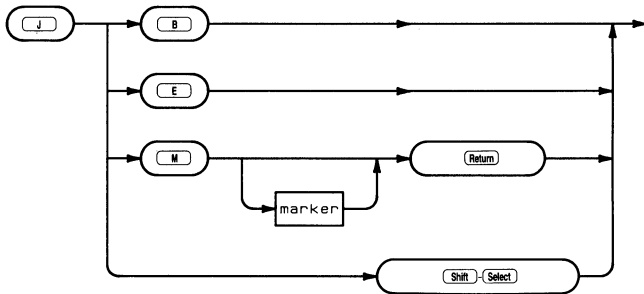
If Auto indent is True, pressing **Return** automatically places the cursor in the same starting column as the previous line. When Auto indent is False, the cursor is positioned according to either the Left or Paragraph margin in the environment.

If Insert is confirmed with **Select** and Filling is True and Auto indent False, all text following the insertion in the same paragraph is margined according to the Right, Left, and Paragraph margin values in the environment. Also, the entire insertion is stored in the copy buffer so you can copy the same text elsewhere if you wish. If Insert is aborted with **Shift** **Select**, regardless of the options set, all changes are aborted and the text and cursor appear as they did when the command was entered.

The Insert command sets the anchor (which can be used by the Zap command) at the position where Insert was initiated. The anchor is set regardless of whether **Select** or **Shift** **Select** is used to exit the command.

Jump

The Jump command repositions the cursor.



Item	Description/Default	Range Restrictions
marker	characters that denote a specific location in a file	1 to 8 ASCII characters, excluding CHR(0) through CHR(31), and CHR(127)

The Jump prompt is:

```
>JUMP: Begin End Marker <sh-sel>
```

The Jump command moves the cursor to the beginning or end of a text file or to a previously defined marker. The command has no other effects; it merely repositions the cursor. To Jump to the beginning of your file, press **J** **B**. To Jump to the end of your file, press **J** **E**.

You can also Jump to a marker by pressing **J** **M**, typing the name of any previously set marker in the file, and pressing **Return**. Marker names are defined with the Set command. A legal marker name is any sequence of up to eight non-control ASCII characters (control characters are deleted by the system). They can actually be longer than this, but the Editor pays attention to only the first eight characters and truncates the rest.


Marker names are not case sensitive. The Editor converts all marker names to uppercase letters so they can be typed using any desired combination of uppercase and lowercase letters. There is a 10-marker limit per text file. See the Set command for more information on markers.

Margin

The Margin command formats all text in the current paragraph to fit the margins set in the environment.



The Margin command provides a means of formatting paragraphs in your file. A paragraph is defined by the Editor to be any text delimited by any combination of blank lines, lines having the Command character as the first non-blank character in a line, or the beginning and end of a text file. (See the Set command for details on the Command character.)


When you initiate Margin (by pressing ) , the Editor takes all the text in the current paragraph (the one where the cursor is) and forces it to fit within the Left, Right, and Paragraph margin boundaries of the environment. After margining, the first line of the paragraph begins at the column specified by the Paragraph margin setting and the rest of the text conforms to the Left and Right margin settings. If a word would exceed the Right margin, it is “wrapped around” or placed on the next line.

Two blanks are allowed following these four characters: ? , : ! . All other blanks are compressed into a single blank character.

Margin is disabled and the system generates an error message unless the environment's Auto indent is False and Filling is True when the command is executed.

Because the Command character in the environment delimits a paragraph, you may want to use it as the first character in each line of tables or lists that you do not want margined.

Note

If a table or list fits the definition of a paragraph, the Margin command will margin that text. Exiting the Insert command with  also uses some of the Margin routine so be aware that these commands can potentially “collapse” a table or list.

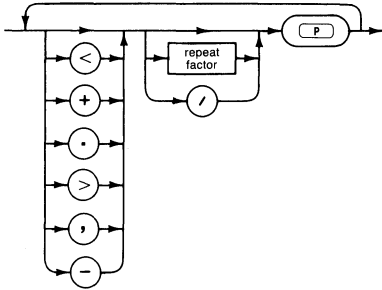
The Margin command has no parameters and its effects cannot be aborted. When writing program text or tables, it is advised that Auto indent be set True, Filling be set False, and the Paragraph margin be equal to the Left margin.

Note

The Margin command clears the contents of the copy buffer regardless of the settings of the Auto indent and Filling options.

Page

The Page command moves the cursor one or more pages or “screens” (23 lines) in the current direction.



Item	Description/Default	Range Restrictions
repeat factor	integer	1 through 1000

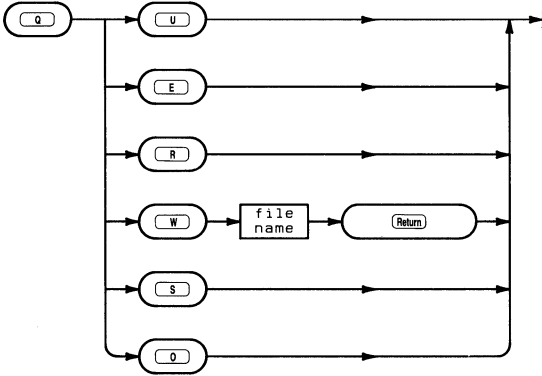
The Page command lets you move rapidly through a text file by repositioning the cursor one or more pages (23 lines of text) forward (>) or backward (<) in a file. Page is executed by pressing , and its movement occurs relative to the position of the cursor. Page moves the cursor in the direction displayed by the Editor prompt when the command is entered. The direction can be changed by pressing > or + for forward, or < or - for backward.

Repeat factors are available in the Page command. For example, to move the cursor three screens or pages in the file, press . The slash character / can be used in place of an integer repeat factor. Pressing moves the cursor to the end of the file (if direction is >) or the beginning of the file (if direction is <). If neither repeat factor nor slash is specified, the default is 1 and the cursor moves one page in the current direction.

and are not available in the Page command. The command is immediately executed when is pressed.

Quit

The Quit command leaves the Editor with various exit options, and returns you either to the main HP EGS Manager menu or to the Pascal command line.



The Quit prompt is:

```

>Quit:
  Update the workfile and leave
  Exit without updating
  Return to the editor without updating
  Write to a file name and return
  Save as file new file EXAMPLE.TEXT
  Overwrite as file EXAMPLE.TEXT
  
```

The Quit command allows you to exit the Editor and store your file on mass storage. The last two quit options shown above are available only if the file existed before the editing session.

Quit is initiated by pressing **Q** from the Editor prompt. Choose any of the options displayed by pressing the first letter of the option.

Pressing **U** (for Update) writes the contents of the text in the computer's memory to a text file on the system volume under the name `WORK.TEXT`. This workfile may or may not be associated with another file name (see the Get and Save commands in the Filer Reference). After writing the file, the system reports the file size (in number of bytes) and displays the main HP EGS Manager menu or the Pascal command line.

Pressing (for Exit) either immediately exits or displays:

```
Are you sure you want to exit without updating?
Type Yes to Exit without update
Type No to Return to Editor
```

This message is displayed only if changes have been made to the text file in the current editing session. If no changes have been made, the system immediately exits when is pressed. If you press , the system exits. Responding with returns you to the Editor.

Pressing (for Return) returns you to the Editor with the cursor located where it was when you Quit.

Pressing (for Write) causes the system to prompt you for a file name. A complete file name is needed. If a volume ID is not given, the default volume is used. If you specify the volume PRINTER:, the file is listed to the system printer.

If you use the Write option and the file already exists, the Editor displays this prompt:

```
>Quit:
FILE.TEXT exists ...
Rewrite then purge old
Overwrite
Purge old then rewrite
None of the above
```

Rewrite then purge old is like the Save command. An attempt is made to write the new file before purging the old.

Overwrite removes the original file and then attempts to write the new version in its place. On SRM units, duplicate links and passwords will be preserved. On a local disc, the file may not fit if the new version is larger than the old.

Purge old then rewrite removes the original file and then attempts to write the new file in the biggest space on the disc. This alternative gives you the best chance that there will be room for the new file.

Whether you Overwrite or Purge the file, the original copy of the file is gone and the only copy of the file is in the Editor's memory. It is advisable to save it on another disc as soon as possible.

None of the above choices return you to the Editor. You may Quit again and write the file with a different name.

Pressing (for Save) writes the file to the original volume and file.

If you try to Save a file and you get the message:

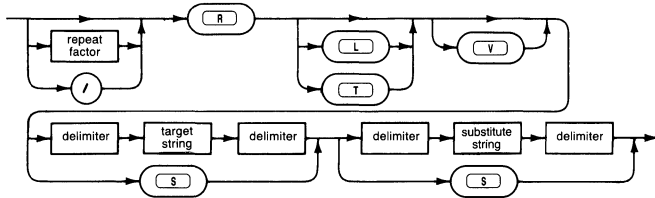
```
>ERROR: No room on vol <space> continues,
```

Press the space bar to continue. You could use another flexible disc with enough space, then Quit, and Save it on the new disc. Alternatively, you could Quit and Overwrite the file.

(for Overwrite) is designed for SRM systems. The Overwrite option allows all duplicate links and passwords to remain accurate. On a local disc, Overwrite may not work if the file has been enlarged. If this happens, press the space bar to continue, Quit, and Save again. The previous Overwrite removed the original file. Now the Save will try to save the file in the largest space on the disc. If this does not work, you must put the file on another disc.

Replace

The Replace command does one or more substitutions of a specified string for another string.



Item	Description/Default	Range Restrictions
repeat factor	integer	1 through 9999
delimiter	character separator that signifies the beginning or end of a string	any non-alphanumeric character; must be used in matched pairs
target string	characters you want to replace	1 through 128 characters
substitute string	characters to substitute in place of the target string	1 through 128 characters

The Replace prompt is:

```
>Repl[1]: L V <tar><sub>=>
```

or

```
>Repl[1]: T V <tar><sub>=>
```

The prompt displayed depends on whether the “Token” definition in the Editor environment is set to True or False. If True, the first prompt is displayed; if False, the second is shown. The Token definitions are explained next.

The Replace command allows you to substitute one string for another in your text file. The anchor (which can be used by the Zap command and accessed by pressing =) is set at the location of the replacement. Replacements can be done to one, all, or only certain occurrences of a string.

In its simplest form, the Replace command is executed by pressing and specifying two strings – a target and substitute – each surrounded by delimiters. The target and substitute strings may be different sizes. Upon typing the final delimiter, the first occurrence of the target string is replaced by the substitute string and the cursor is positioned at the end of the substitution.

A target string (the one that you want replaced) must be supplied. A string is a contiguous series of non-control ASCII characters surrounded by delimiters. Delimiters signify the beginning and end of a string and are characters such as: / ' , and \.

A substitute string (what you want the target string changed to) must also be supplied with delimiters. The substitute may be an empty (null) string.

Don't use a delimiter that appears within your string. Delimiters must be matched pairs; if you use \$ to signify the beginning of a string, you must use \$ to signify its end. The substitute string can have a different set of delimiters than the target string and the two strings may be of different sizes. The maximum length of either string is 128 characters.

After pressing , the prompt on your screen contains either an L or T for “literal” or “token” modes. Literal and token are mutually exclusive; if one option is shown as available, the other is automatically the default. If is shown in the prompt and you want to use the token search mode, then type in the two strings and their delimiters. The replacement takes place in the default mode, in this case, token. To do the same replacement in the literal mode, press and type in both strings as before. The Replace command then searches for a literal form of the string.

A literal string is exactly that – a literal string of characters either isolated or embedded in a word. A token string is usually a word (a string isolated by delimiters). Delimiters in this context are any ASCII characters except numbers or alphabetic characters – they do not have to be matched pairs. Blanks, commas, and periods are the most common delimiters in English text because they separate words.

To illustrate literal and token replacements, the following example assumes the direction is forward (>) with the cursor located at or before the start of the sample sentence. In the sentence

```
That's my hat!
```

a token replacement for `hat` with `umbrella` replaces the last word `hat` in the sentence with `umbrella`. In contrast, literal replacement would substitute `umbrella` for the `hat` imbedded in `That's`, resulting in

```
Tumbrella's my hat!
```

Direction applies in the Replace command and is shown by the first character in the command's prompt. If the direction is forward (>), the replacement occurs between the cursor position and the end of the file; if backward (<), the replacement occurs between the cursor and the beginning of the file.

Repeat factors are available for the Replace command but must be typed before the command is initiated (before is pressed). A repeat factor causes that number of substitutions to be made. If not specified, the repeat factor defaults to 1. A slash character (/) may also be used to change all occurrences of the specified string in the current direction. The repeat factor (or slash character) is displayed in brackets ([]) in the command's prompt. The repeat factor works differently when the Verify option is used.

The Verify option lets you choose whether or not to make a particular replacement. The combination of a repeat factor with Verify allows you to replace only certain occurrences of a string in the file. For example, after you press and type in the target and substitute strings, the Editor moves the cursor to the first occurrence of the target string and prompts:

```
>R[1][2]: <sh-exc> aborts,R replaces,' ' doesn't
```

To confirm the replacement, press . To skip to the next replacement (if any), press the space bar. While using Verify, pressing aborts the operation and retains all replacements made up to that time.

The Same option is available with Replace and refers to either the most recent target string (used in a Find or Replace) or the most recent substitute string (used only in Replace). Which string it signifies (target or substitute) depends on where it is used in the Replace command. To use Same, simply press **S** in place of the delimited string. If you type **S** followed by a delimited string, the most recent target is replaced with the specified string. If you type a delimited string followed by **S**, the specified target is replaced with the last substitute. Both strings may be specified by typing **S S**. The current assignments of Same patterns can be seen by pressing **S E** (see the Set command for more details).

Note

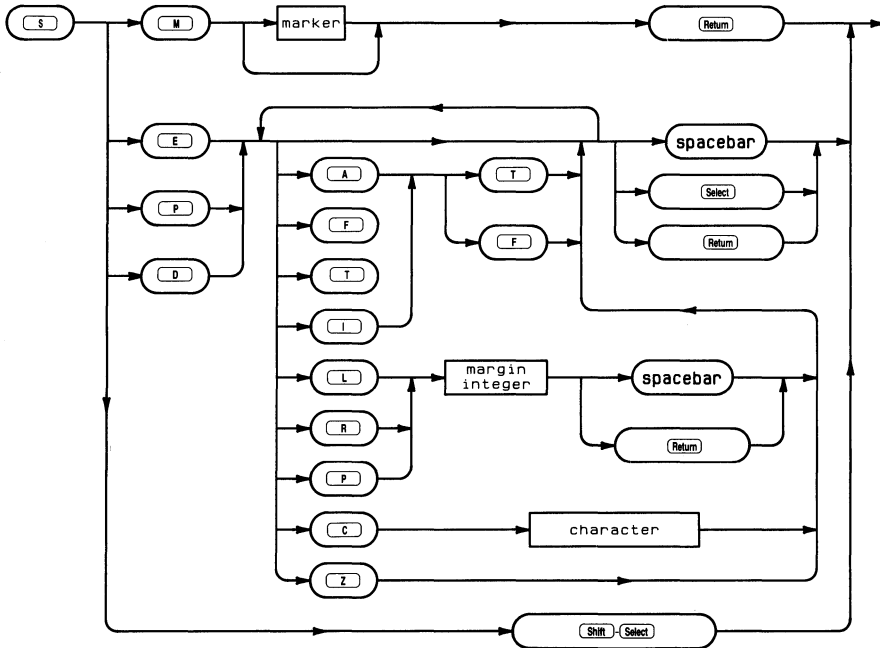
If a Find has been done since the last Replace, the target string used by the Same option is now the target specified in the Find command.

The Ignore case option applies to the Replace command. Type **S** and **E** to Set the Environment. Type **I** and **T** to set Ignore case to True. Type **T** and **T** if Token is not already True. The target string and all token strings in the text are treated as uppercase. When a match is found, the token string is replaced with the substitute string. The case of the substitute string is not affected by the Ignore case option.

The Replace command can be aborted before all specifications are complete by pressing **Shift Select**. (Subsequent use of the Same option after aborting the Replace command may give you unwelcome results.)

Set

The Set command defines markers and alters the environment in which your text operations occur.



Item	Description/Default	Range Restrictions
marker	characters that denote a specific location in a file	1 to 8 ASCII characters, excluding CHR(0) through CHR(31), and CHR(127)
margin integer	integer	0 through 9999; left margin must be less than right margin; recommended use is any integer less than the screen width
character	any printable character	any valid ASCII character; recommended values are between CHR(32) and CHR(126)

The Set prompt is:

```
>Set: Env Mkr Prog Doc <sh-sel>
```

Set lets you define markers and various environment parameters. Markers are Set by moving the cursor to where you want the marker, pressing **S** **M** (for Set Marker), and typing in a marker name followed by **Return**. A marker name is any sequence of up to eight non-control ASCII characters. The Editor accepts more than eight characters but truncates anything longer. All non-printing characters (those with an ASCII value of either 127 or in the range of 0 to 31) are deleted by the system. The Editor converts these names to uppercase so they can be typed in whatever form is convenient.

No more than 10 markers can be set in a file. If you try to set more than 10, the Editor displays the markers in a numbered list and prompts you for the number of the marker you wish to replace. All markers can be removed by using the Zap markers command from the Environment. Markers are used with the Jump and Copy commands, and their names are shown in the environment display. The locations of the markers are not shown, so the use of meaningful marker names is advised.

Pressing **S** **E** (for Set Environment) displays the current environment and allows you to change the parameters of the environment. When entering the Editor with a new file, the default environment is the Program environment which looks like this:

```
>Environment: {options} <sel> or <sp> leaves
  Auto indent   True
  Filling       False
  Left margin   0
  Right margin  78
  Para margin   5
  Command ch    ^
  Token def     True
  Ignore case   False
  Zap markers
  275 bytes used, 348909 available.
```

```
Patterns:
  <target>= 'inery', <subst>= 'inary'
```

```
Markers:
  TOP          FIX
```

```
File EXAMPLE.TEXT
Date Created: 7-10-85  Last Used: 7-10-85
```

Patterns and Markers are shown only if they have been set. The heading near the bottom displays a file name if the Editor is entered with a specified file. Whenever a TEXT file is saved on mass storage, the current environment is saved with it and becomes the default environment when that file is used by the Editor. (Note that the environment is not saved with Data and ASCII files.)

The environment also displays how many bytes of memory have been used and how many are still available for use in the Editor. The total number of bytes (used and available) depends on the amount of memory in your machine.

To change a parameter in the environment, press the first letter in the parameter's name. The cursor is automatically positioned at the item to be changed and the new value must be typed. If the parameter needs a number (as in Left, Right, and Paragraph margins), then the number must be followed by pressing **Return** or the space bar. All other parameters accept a single character and return the cursor to the environment's prompt as soon as the character key is pressed.

Automatic indenting is a Boolean value (either true or false) that affects the Insert and Margin commands. When inserting text with this item set true, pressing **Return** automatically moves the cursor to the next line at the same starting column as the previous line. This indenting feature is useful when writing unmarginized text (like programs), so it is set true for the Program (default) environment.

When Auto indent is true the Margin command is disabled. When Auto indent is false, pressing **Return** places the cursor on the next line at either the Left margin or Paragraph margin (as currently defined in the environment).

Filling is another Boolean value that affects the Insert and Margin commands. It usually has a value opposite that of Auto indent. When set True (and auto-indent is False), filling causes automatic “wrap around” of text. If a word is too long to fit on the current line (as defined by the Right margin value), it is carried or wrapped around to the next line and no **Return** is necessary. Another effect of this parameter being set true is that an Insert completed by pressing **Select** causes all text following the insertion in that paragraph to be margined or filled according to the current values of the Left, Right, and Paragraph margin settings. All blanks in the text are then compressed to a single blank (though two blanks are allowed following the four characters: ? , ! :). The Margin command only works when Filling is set True and Auto indent is set False.

With Filling set False, the wrap around and margining functions are disabled. When approaching the end of a line, the system generates a “beep” to inform you that you need to press **Return** to go to the next line. If you type past the display area of the screen, an exclamation point (!) is shown in the last column. The text, though not visible, is maintained in the computer’s memory.

The Left margin may be set to any integer between 0 and 9999. Numbers longer than 4 digits are truncated by the system. The Left margin must be less than the Right margin setting or an error message is generated when you attempt to exit the environment.

The Right margin setting has the same numerical limitations as the Left margin. Unless you have a particular reason for doing so (like making full use of a 132-column printer), it is not a good idea to set this margin beyond the right column display limits of your screen because the text will not be visible.

The Paragraph margin can be set to any positive integer up to 4 digits. This setting determines the indentation that the first line in each paragraph will get. This occurs when Filling is set false (while inserting text) or when Margin is used. Note that a paragraph as defined by the Editor is any text surrounded by blank lines or by lines beginning with the Command character. The beginning and end of a file will also delimit paragraphs.

The Command character can be any non-control (or printable) ASCII character. If this character is the first non-blank character in a line, the Margin command treats the line as if it were blank. The line is not margined and it is considered to be the beginning or the end of a paragraph. The default Command character is the `^` character.

Token is used by the Find and Replace commands. When Token is set True, the default value for Find or Replace becomes token and the command's prompt displays the literal option. (Token and literal refer to the type of target string searches that take place in these commands.) Conversely, if Token is False, the default value for Find and Replace is literal and the command's prompt displays Token as an option.

The Ignore case option affects searches in the Find and Replace commands. If Ignore case is False, then "string" and "STRING" and "String" are not treated as equal. If Ignore case is True, they are treated as equal. This only works when the Environment's Token mode is True or if you type a T before typing the target string.

The Zap markers command removes all markers from the file.

The environment display is exited and the Editor's main prompt returned by pressing **Return**, **Select**, or the space bar. The current environment settings are saved in the mass storage file only if it is of type TEXT.

Although there is only a single environment associated with a text file, the environment may be set to one of two predefined configurations: the Program environment (by pressing **S** **P**) and the Document environment (by pressing **S** **D**). These configurations optimize the various environment parameters for writing programs or regular (non-program and non-tabular) text, respectively. When either predefined environment is Set, the current environment is displayed and any of its parameters can be changed. If you want to change just one or two parameters, use **S** **E** to get into the existing environment.

Changes made to the environment cannot be aborted but the parameters may be changed as many times as desired.

Verify

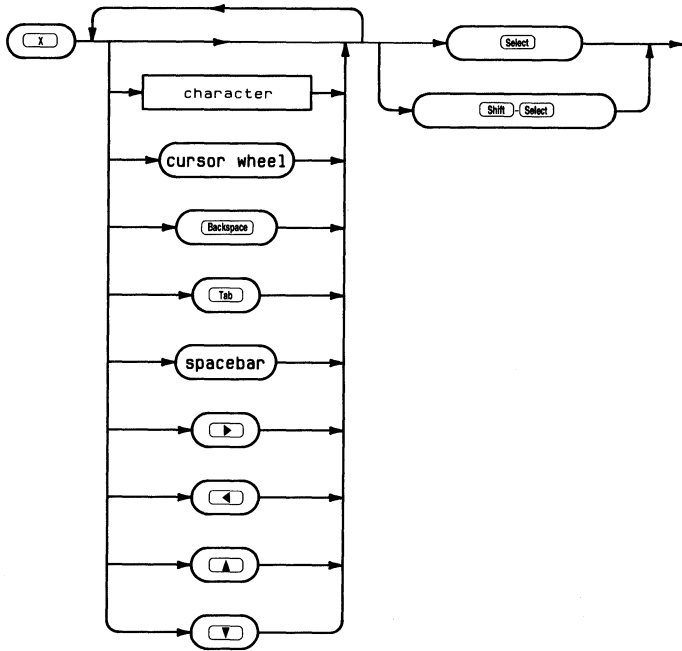
The Verify command refreshes the screen display from memory.



The Verify command has no options; it is executed immediately by pressing **V**. Verify causes the Editor to refresh or update the current screen display from memory, move the current line (the one where the cursor is) to the middle of the screen, and display the Editor prompt. If the cursor is located in the first 23 lines of text when Verify is used, the line containing the cursor is not moved.

eXchange


The eXchange command replaces text character for character at the cursor position.



Item	Description/Default	Range Restrictions
character	any printable character	any valid ASCII character; recommended values are between CHR(32) and CHR(126)

The eXchange prompt is:

```
Xchng: Text <bs> <sh-sel> aborts <sel> accepts
```

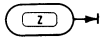
The eXchange command enables you to exchange text characters one for one at the cursor position. eXchange operates only on the current line (the line where the cursor is located when the command is entered). **Backspace**, , and the cursor wheel (in backspace direction only) move the cursor one space back and display the character that was replaced.

Although any ASCII character can be used in eXchange, use of control characters is not advised. Carriage returns cannot be entered since the command is unable to cross line boundaries. Direction and repeat factors do not apply to the eXchange command. Any cursor control key that does not effectively “backspace” the cursor will generate question marks while executing eXchange. Backspacing past the point at which eXchange was entered is not allowed.

eXchange is initiated by pressing **X** and is exited by pressing **Select** or **Shift** **Select**. **Select** confirms the exchanges, returns the Editor prompt, and displays the cursor at its position when **Select** was pressed. **Shift** **Select** returns the copy of the text file in the computer’s memory to its state before eXchange was entered, displays the Editor prompt, and shows the cursor at its position when eXchange was entered.

Zap

The Zap command deletes text and fills the copy buffer with the deleted text.



The Zap command has no options; it is executed immediately by pressing **Z**. Zap deletes all text between the anchor and the current cursor position and stores it in the copy buffer. The anchor is located at the position in the text where the most recent Adjust, Find, Insert, or Replace command was executed. You can confirm the position of the anchor by pressing **=**, which moves the cursor to the anchor.

If more than 80 characters are going to be Zapped, the Editor displays a prompt asking if you wish to Zap anyway. Also, if the Copy buffer is not large enough to store the deletion, a prompt asks if you wish to go ahead and Zap the text. (Use the Set environment command to see how much memory is available; the copy buffer shares this memory with that used to hold the text file in memory.)

You can recover the deleted text the Copy Buffer command. Zap can also be used to move large chunks of text from one location to another within a file.

Note that the effects of Zap can be surprising since the anchor position is set by four different and commonly used commands. Therefore, it is a good practice to check the location of the anchor before executing a Zap.

Filer Reference

Chapter 6: Filer Reference

Information Provided.....	125
Filer Command Quick Reference.....	126
File and Volume Specifications.....	127
Access.....	128
Bad sector.....	130
Change.....	131
Duplicate link.....	133
Extended directory.....	135
Filecopy.....	138
Get.....	141
Krunch.....	142
List directory.....	144
Make.....	146
New.....	149
Prefix.....	150
Quit.....	152
Remove.....	153
Save.....	155
Translate.....	157
Unit directory.....	159
Volumes.....	161
What.....	162
Zero.....	163

Filer Reference

Chapter

6

Information Provided

This section describes the Filer commands available from the HP EGS operating system. It contains a quick reference for the commands and an alphabetized description of the command syntax and options. For information on interpreting the syntax diagrams, see the introduction to the *HP EGS Syntax Reference*.

Filer Command Quick Reference

Command	Description
Access	Changes the access rights (via passwords) on a file or directory (SRM only).
Bad sector	Scans a volume and searches for unreliable (bad) storage areas (local disc only).
Change	Changes the name of a file, set of files, or volume.
Duplicate link	Duplicates links to a file or set of files (SRM only).
Extended directory	Lists complete directory information about a specified volume or set of files.
Filecopy	Copies a file, set of files, or a volume to a specified destination.
Get	Specifies a file as the workfile.
Krunch	Consolidates all unused space on a volume in a single area by packing the existing files together (not valid for SRM).
List directory	Lists partial directory information about a specified volume or set of files.
Make	Creates a directory (for SRM) or a file on a volume.
New	Specifies that no file is the current workfile.
Prefix	Specifies a new default volume.
Quit	Provides an orderly exit from the Filer.
Remove	Removes a file or directory.
Save	Saves the current workfile(s) with the specified name.
Translate	Translates text files of types TEXT, ASCII, and Data to other text file representations or to unblocked volumes.
Unit directory	Sets the default unit directory (SRM only).
Volumes	Lists the volumes currently on-line.
What	Lists the name and current state (saved or not saved) of the workfile(s).
Zero	Creates an empty directory on the specified volume (not valid for SRM).

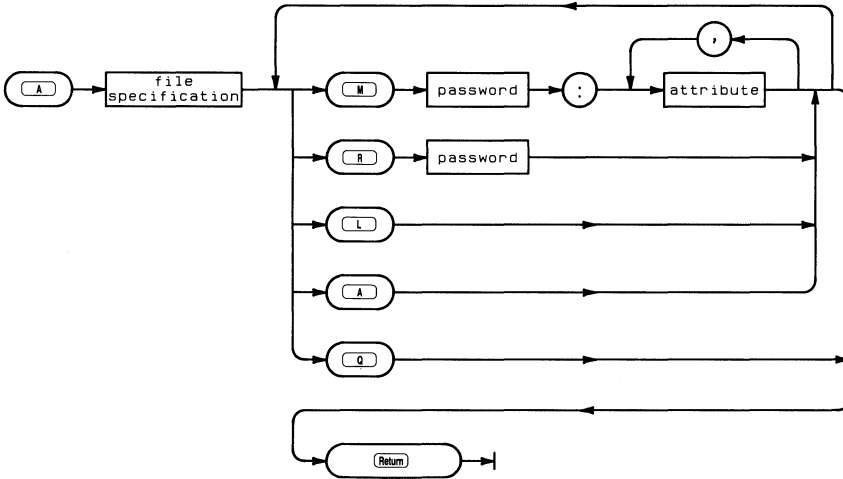
File and Volume Specifications

Several of the syntax diagrams on the following pages refer to the “volume specification” and the “file specification”. The “volume specification” is the syntax for commands that operate on volumes. The “file specification” is the syntax for commands that operate on files. Volume specifications need not be followed by a colon (:) except when a literal volume name is given. Then the name must end with : to distinguish it from a file name. If no volume specification is given, the default volume is assumed.

For more information on file and volume specifications, see *Files, Volumes, and Mass Storage* in this manual.

Access

The Access command allows you to change public access rights on SRM files. This command does not work on local disc files.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	a legal SRM file specification
attribute	characteristic assigned to a password	MANAGER READ WRITE SEARCH PURGELINK CREATELINK ALL
password	word that secures access to a volume or file	up to 16 characters, excluding > : .

The Access prompt is:

```
Access rights for what file ?
```

All access capabilities for a file are initially public. You can remove one or more capabilities from public access by associating them with password(s).

To answer the prompt, you should type the file specification. If the file already has a MANAGER password, then you must include the password in the file specification.

The next prompt is:

```
Access: List, Make, Remove, Attributes, Quit ?
```

These are the possibilities. You can list the attribute passwords, make new ones, or remove passwords. The Attributes option just lists the possible attributes for your help. Quit returns you to the Filer prompt.

Access rights cannot be changed on open files or open working directories.

To make new passwords, press . You see this prompt:

```
Make password:attribute ?
```

Type the password (up to 16 characters), then a colon (:), followed by the attribute list (with attributes separated by commas). Different passwords may be associated with each attribute or one with ALL. If you type a password that already exists, you are asked:

```
PASSWORD already exists...replace it ? (Y/N)
```

Note that passwords should not contain these three characters: > : , .

To remove passwords, press . You see the prompt:

```
Remove password ?
```

When you type only the password, all attributes associated with it are cleared.

The Attributes option list is:

```
MANAGER
READ
WRITE
SEARCH
PURGELINK
CREATELINK
ALL
```

Bad sector

The Bad sector command scans a mass storage medium for errors (not valid for SRM).



Item	Description/Default	Range Restrictions
volume specification	complete volume name	any legal volume specification

The Bad sector prompt is:

```
Bad sector scan of what directory ?
```

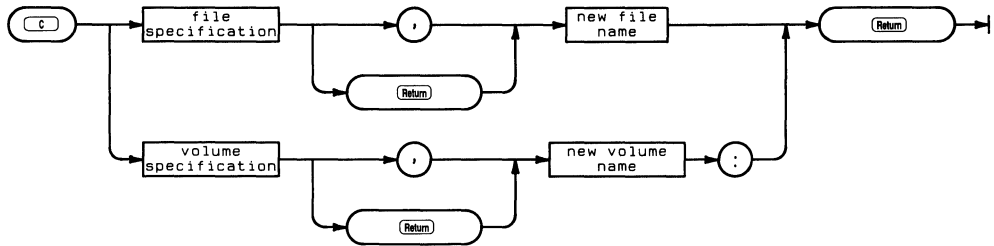
The Bad sector command allows you to check a mass storage medium to find out if each block (sector) is readable. Mass storage media may become unreliable after damage or excessive wear.

Press **B** to initiate the command and answer the prompt with a volume specification. The Filer then displays a message indicating that it is scanning the volume from block 0 to the end of the volume. The Filer does a read operation on each sector and does a cyclic redundancy check (CRC) on the data. If the CRC results are normal, then that sector is considered to be good; if not, the Filer lists the sector number.

If you find a bad sector in a file, you may wish to use the Filer to change the file type (suffix) to `.BAD`. The `BAD` file will not be moved in a Krunch operation. A large number of bad sectors indicates worn-out media. The worn out media should be used only if you are willing to risk losing information on that volume.

Change

The Change command lets you rename files, directories, and volumes. This command performs the same function as the CHANGE command available in the HP EGS Graphics Editor.



Item	Description/Default	Range Restrictions
file specification	volume and file name you want to change	
volume specification	volume name to change	
new file name	name that replaces old file name	any valid file name
new volume name	name that replaces old volume name	(see the beginning of this section)

The Change prompt is:

```
Change what file?
```

The Change command requires two specifications: the original volume or file specification and the new one. The two specifications can be separated by either a comma or a carriage return. If you are changing the name of a volume, any volume specification (that is legal for the volume's Directory Access Method) can be used for both specifications.

To change the name of a file, use any legal volume ID in the first specification and only the new file name in the second specification. The Filer is intelligent enough to know that the file whose name you are changing resides on the volume identified in the first specification. After the Filer has finished changing the name(s) and updating the directory, it reports the name change(s) on the CONSOLE: volume.

Because many of the Pascal subsystems append the string ,TEXT or ,CODE to a file name given in response to a prompt, it is a good idea to retain these parts of a file name when making a change.

Wildcards (the * and ? characters) may be used in the Change command. If a wildcard is used in the first specification, it must also be used in the second one. The subset string that is replaced by the wildcard in the second specification (the new name) is the same as the string it stands for in the first specification.

Suppose you have a volume named BUGS: with the following files:

```
WHATISIT.TEXT
WHOISIT.TEXT
WHYISIT.TEXT
```

Specifying BUGS:WH=TEXT, FO=FA in response to the Change prompt results in the following messages being reported by the Filer:

```
BUGS:WHATISIT.TEXT  changed to FOATISIT,FA
BUGS:WHOISIT.TEXT  changed to FOOISIT,FA
BUGS:WHYISIT.TEXT  changed to FOYISIT,FA
```

Here is another example using the files shown above on the BUGS: volume. Specifying BUGS:WH=,TEXT, = results in:

```
BUGS:WHATISIT.TEXT  changed to ATISIT
BUGS:WHOISIT.TEXT  changed to OISIT
BUGS:WHYISIT.TEXT  changed to YISIT
```

You may wish to create some empty files using the Make command and experiment with them before using wildcards extensively. Until you get used to them, the effects of wildcards are not always obvious.

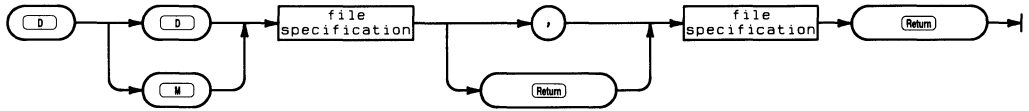
Note

Using the Change command to “change” a file name to the same name results in the file being removed.

The Change command does *not* change the file type.

Duplicate link

The Duplicate link command establishes a new pointer to a file (SRM only).



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	

The Duplicate link prompt is:

```
Duplicate link (valid only for SRM type units)
Duplicate or Move ? (D/M)
```

Do you want the original pointer to the file removed after the duplicate link is established? If you do, type M for Move; if not, type D for Duplicate.

If you choose the Duplicate option, the next prompt is as follows:

```
DUP_link what file ?
```

Type the SRM file specification (including the password if the CREATELINK capability has one).

```
DUP_link to what ?
```

Type the new file specification. Wildcards can be used in the specification. This puts a link to the file in a second directory.

If the file is referenced from two or more directories, the file is physically removed from the disc only when all links to the file have been removed.

You should be aware that new CODE files generated by the Compiler, Assembler, and Librarian to replace older versions are not written in the same space (unless Overwritten). If several directories have duplicate links to the same CODE file and the CODE file is recompiled, only one directory has an accurate link to the *new* CODE file. Other users must use the Duplicate link command to become linked to the new CODE file.

Note

Using the Duplicate command to “duplicate link” a file to the same file results in the file being removed.

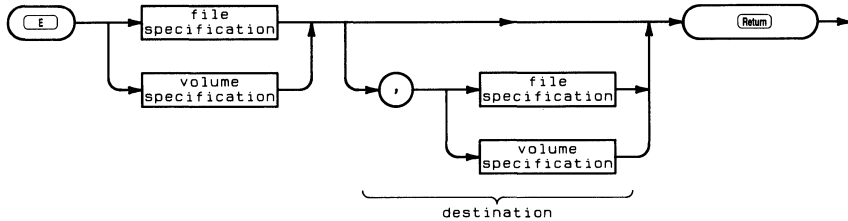
If you choose the Move option, the next prompt is as follows:

```
Move what file?
```

Type the new file specification. Wildcards can be used in the specification. This puts a link to the file in a second directory, and the original link is then removed.

Extended directory

The Extended directory command lists the directory of a blocked volume or a set of files in the volume.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	
volume specification	complete volume name	

The Extended directory prompt is:

```
List_ext what directory ?
```

The Extended directory command requires a legal volume or file specification. Results can be listed to the PRINTER: or to a text file if specified and separated from the first specification by a comma. If no destination is specified, the listing defaults to CONSOLE:. Wildcards are available to identify subsets of files on the volume.

In the listing, the name of the volume is displayed in the upper left corner. To the right, the directory type is displayed. Pascal LIF discs have Level 1 directories; level 1 directories contain the creation date and volume size information. Level 0 directories (created on other systems) do not. Your directory listing should display the date the directory was created and the date it was changed, the size of the storage blocks, and whether the listing is in Storage order or Alphabetic order. Files are normally listed in storage order. To list them in alphabetical order, include [*] after the directory name. For example:

```
MYDIR:[*]
```

The size of blocks on LIF volumes is 256 bytes. The size of blocks on WS1.0 volumes is 512 bytes. The Shared Resource Management system uses single-byte “blocks”.

The column entries for each file include: file name, number of blocks used for storage, the file size in bytes, the number of the block where the file starts, the date the file was changed, the type as recognized by the file system, the type-code used by the directory system, SRM access information, the date the file was created and two extension fields.

The SRM access information column comes under the heading “directory info”. It contains codes which show the access rights which are still public (that is, they have not been protected by associating them with a password):

M	Manager
R	Read
W	Write
S	Search
P	Purgelink
C	Createlink

It also shows the current file status:

CLOSED
SHARED
EXCLUSIVE
CORRUPT

CLOSED, SHARED, and EXCLUSIVE are file statuses associated with SRM systems. If a file is ever marked CORRUPT, stop your operation and notify the person responsible for your SRM. (This person should restore the SRM to a usable state.)

The last two lines display additional directory information including how many more entries can fit in the directory. (The available number of entries in a directory was specified when the disc was initialized.)

The results can be listed to a printer or a file. The destination of the listing is separated from the volume specification or file specification being listed by a comma. If no destination is specified, then the listing defaults to the screen. Wildcards are available to specify groups or subsets of files on a mass storage medium.

For example, assuming that SYSVOL: (the system volume) is in the disc drive that has been assigned logical unit #3, a listing of all the CODE files on that volume could be sent to the printer by specifying any of the following in response to the Extended directory prompt:

- #3:=CODE,#6: Specifies to list the volume residing in unit #3 to logical unit #6: (the PRINTER: volume).
- *=CODE,PRINTER: Specifies to list the system volume to the PRINTER. (Without the colon, the listing would be sent to a DATA file named "PRINTER" on the default volume.)
- SYSVOL:=CODE,#6 Specifies to list the SYSVOL: volume to unit #6.

In all cases the =CODE string refers to all files whose names end in CODE on the specified volume and the listing is sent to the printer.

Listings can also be sent to a file. Use a destination parameter after the source parameter (separated by a ",") as in the above PRINTER: example. Give a complete file specification, using the appropriate suffix in the file name. Otherwise, a file of type Data is produced. For example:

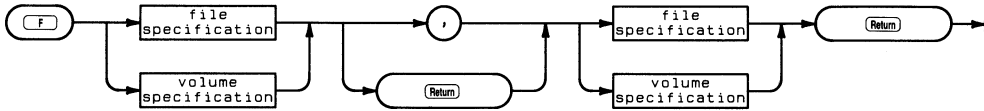
```
List what directory ? #3:,SYSVOL:LIST,TEXT
```

or

```
List what directory ? #3:,SYSVOL:LIST,ASC
```

Filecopy

The Filecopy command copies a specified file, set of files, or volume to the specified destination. This command performs the same function as the FILECOPY command available in the HP EGS Graphics Editor.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	
volume specification	complete volume name	

The Filecopy prompt is:

```
Filecopy what file ?
```

The Filecopy command is initiated by pressing **F** and requires two specifications – a source and a destination – separated by either a comma or **Return**. The source volume must be on-line. The destination volume does not have to be on-line (which allows you to copy files on different volumes using a single-volume system).

Copying Files

Wildcards may be used to specify sets of files. If the equals wildcard (=) is used, the copy is not confirmed before taking place. Also, note that if the equals wildcard is used alone (without any qualifying strings) then the Filer copies every file on the specified volume. If the question mark wildcard (?) is used, you are asked to confirm the copy of each file meeting the wildcard specification before the Filecopy takes place. Thus, using the ? wildcard allows you more flexibility and control over the process.

The dollar sign wildcard (\$) may be used in the destination specification to indicate that the file(s) will have the same name (or names) as the source file(s). For example, assume that there are a number of TEXT files on the volume DRAW: and that a second volume named MINE: exists. The following specifications:

```
DRAW:=TEXT,MINE:$
```

result in all the files on the DRAW: volume whose file names end with the string TEXT being copied to the volume MINE: and given the same name as they have on the DRAW: volume.

When copying only files, be sure to use either a file name or the \$ character when specifying a destination. If in the above example the destination was specified as MINE: instead of MINE:\$, the Filer would respond:

```
Destroy directory of MINE: ?
```

If you respond with , the directory of that volume gets overwritten. Pressing aborts the Filecopy command and returns the Filer prompt.

On a system with a single mass storage volume, the Filecopy command proceeds by reading the specified file(s) into memory, prompting you to remove that volume and insert the destination volume, and then writing the file(s) in memory onto the destination volume. Depending on the amount of memory in your computer and the amount of material being copied, you may have to swap discs more than once.

Note

When using the Filecopy command with a single mass storage volume, wait for the Filer's prompt before removing the source volume and replacing it with the destination volume. Failure to follow this guideline may result in the loss of information from the source volume.

A size specification may be used in the destination file specification. For example, specifying:

```
SYSVOL:FILE,OTHERVOL:FILE[35]
```

would result in the file being written to the first available area on OTHERVOL: that was at least 35 blocks in size.

Copying Volumes

You can also use the Filecopy command to make a back-up copy of an entire volume. Simply type in the source volume specification and the destination volume specification. The destination volume must have been initialized, but it does not have to have been Zeroed (because the directory gets copied from the source volume). The Filer will ask you if you want the directory destroyed. A volume-to-volume copy makes an exact copy of the source volume on the destination volume.

Note that having two volumes with the same name on-line at one time is not advised. The Filer looks for volumes according to their volume names and may not be able to distinguish one from the other. Thus, the Filer may perform an action on one volume when you wanted the operation to affect the other volume. The Filer warns you whenever this condition exists. If you get a warning, either remove one of the volumes or use the Change command from the Filer to change the name of one of the volumes.

You can copy files on one volume to a volume of a different size, but you should not use volume specifications alone to do this. If the source volume is larger than the destination volume, the Filer refuses to execute the Filecopy. If the source is smaller than the destination, the destination volume ends up the same size as the source when the operation is through, so you lose storage space. (Remember that it makes an exact duplicate of the source.)

The best way to handle copies between different-sized volumes is to use one of the wildcards. Use the equals wildcard (=) if the destination is larger than the source, and the question mark wildcard (?) if the destination is smaller than the source. In the latter case you may have to be selective in your copies, because there may not be enough space for all of the files.

When the Filecopy command has finished its task, the screen displays what file(s) or volume has been copied, and the Filer prompt appears. The Filecopy command can be aborted before all specifications are complete by pressing **Return** in response to the prompt.

In cases where the destination volume already contains a file with the same name as the file being copied, this prompt is displayed:

```
ANYVOL:XFILE
exists ... Remove, Overwrite, Neither ? (R/O/N)
```

You have the options:

- **Remove:** Remove the file before proceeding with the operation.
- **Overwrite:** Replace the contents of the old file with the new information. The Overwrite option cannot be used to change the type of a file on SRM. Attempting to do so will result in the file contents being inconsistent with the file type.
- **Neither:** Cancel the operation.

The Overwrite option allows you to put a file in the same starting location as the original. This is important to SRM users when duplicate links exist to a file. All links and passwords to the file are accurate when a file is updated with Overwrite, because it is put in the same logical location. If you chose the Remove option, the original file would not actually be removed; only your link to it is removed. The other users are still linked to the original file.

Note

Using the Filecopy command to “copy” a file name to the same name on the same volume results in the file being removed.

The Filecopy command does *not* change the file type.

Overwrite of a file of type SYSTM is not recommended, because the start execution address cannot be changed in an existing SYSTM file.

Get

The Get command associates a specified file as the current workfile.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	

The Get prompt is:

```
Get what file ?
```

The Get command is initiated by pressing **G** and it prompts you for a file specification. If a workfile currently exists when the Get command is executed, you are asked if you want to release that file before being allowed to specify a new workfile. Upon receiving the specification, the Filer finds the file(s) and associates that name with the current workfile. Subsequent operations on the workfile use the specified name. The workfile is generally *WORK.TEXT and/or *WORK.CODE.

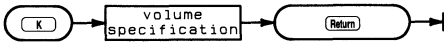
The Get operation assumes that the text version of the specified file has a .TEXT suffix. If the text version is ASCII, you must include the .ASC suffix. If the text version is Data, you must include a period (.) at the end of the file name to prevent the appending of the .TEXT suffix.

The operating system notes that either a TEXT or CODE or both versions of the workfile exist. Workfiles can only be of type .TEXT/.ASC/Data or of type CODE. If both text and code versions of the specified file exist, both are associated with the workfile; if only one exists, the association is made with that file. The Filer reports one of three things: either a text or code file has been loaded, both have been loaded, or the file cannot be found on the specified volume.

The Filer is not the only Pascal subsystem where a workfile can be created. The Editor, Compiler, and Assembler subsystems also create workfiles. Once a workfile exists, it is treated as the default file in many of the subsystems. A workfile is “released” by the Filer’s New command.

Krunch

The Krunch command moves all files on a block-structured volume so that all the unused storage space is at the end of the volume. This command performs the same function as the PACK command available in the HP EGS Graphics Editor.



Item	Description/Default	Range Restrictions
volume specification	complete volume name	

The Krunch prompt is:

```
Crunch what directory ?
```

The Krunch command is initiated by pressing and it prompts you for a volume specification. After you respond with a legal volume specification of an on-line, block-structured volume, it prompts:

```
Crunch directory MKWORK: ? (Y/N)
```

Where MKWORK: is whatever volume you specified. Typing for Yes lets the command continue; for No returns the Filer prompt. The Krunch command executes a sensitive operation – that of moving all the files forward on the disc by reading the files into memory and then writing them back out on the disc in such a manner so as to make all the unused space on the volume contiguous at the end of the disc.

If there is the slightest question about the reliability of the mass storage medium you are using (because of excessive wear or damage), use the Bad sector command to do a scan of the sectors on the volume before initiating Krunch. If a bad sector is found, use the Make command from the Filer to make a file of type .BAD over the bad sectors. Krunch does not move files of type .BAD. Moving files onto an unreliable area of storage is a good way to lose them.

CAUTION

UNDER NO CIRCUMSTANCES SHOULD YOU ATTEMPT TO INTERRUPT THE KRUNCH OPERATION ONCE IT HAS BEGUN. YOU ARE RISKING YOUR DIRECTORY, AND THUS ALL OF THE INFORMATION CONTAINED ON THAT VOLUME IF YOU DO SO. DO NOT TOUCH THE POWER SWITCH OR THE DOOR ON THE DISC DRIVE, OR ATTEMPT TO USE THE KEYBOARD WHILE A KRUNCH IS IN PROGRESS.

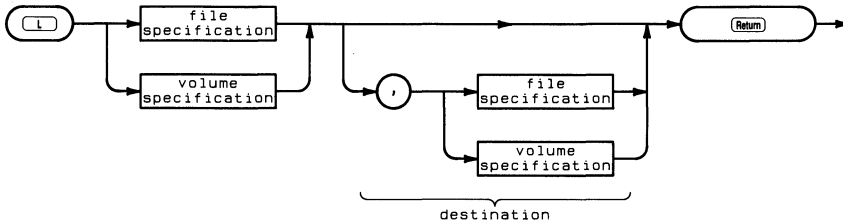
This process becomes necessary when, after repetitive reading and writing to the disc, the available storage space becomes highly fragmented. For instance, suppose you have 100 blocks available on the disc, but because they are all in 10- or 15-block chunks, there is not enough contiguous storage space for the system to write a 20-block file to the disc.

The Krunch command is extremely useful and using it should not worry you. However, because it alters the directory (which maps where the information on the disc resides), it is one of the quickest ways to wipe out a volume. The precautions outlined above should help you avoid any problems while using the command.

The Krunch command does nothing on SRM units.

List directory

The List directory command lists directory information about a block-structured volume or one of its subsets. This command performs the same function as the CATALOG command available in the HP EGS Graphics Editor.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	
volume specification	complete volume name	

The List directory prompt is:

```
List what directory ?
```

The List directory command requires a legal volume or file specification. Results can be listed to the PRINTER: or to a text file if specified and separated from the first specification by a comma. If no destination is specified the listing defaults to CONSOLE:. Wildcards are available to identify subsets of files on the volume.

In the listing, the name of the volume is displayed in the upper left corner. To the right, the directory type is displayed. Pascal LIF discs have Level 1 directories. Level 1 directories contain directory-create and volume size information. Level 0 directories (created on other systems) do not. Your directory listing should display the date the directory was created and the date it was changed as system volume, the size of the storage blocks, and whether the listing is in storage order or alphabetical order. The size of blocks on LIF volumes is 256 bytes. The size of blocks on WS1.0 volumes is 512 bytes. The Shared Resource Management system uses single-byte "blocks".

To have directories listed in alphabetical order, include [*] after the directory name. For example:

```
MYDIR:[*]
```

The column entries for each file include: file name, number of blocks used for storage, the file size in bytes, and the date the file was created or changed.

The last two lines display additional directory information including how many more entries can fit in the directory. The size of a directory is specified when the disc is initialized. You need one 256-byte block for each eight directory entries.

For example, initiating the command by pressing **L**, specifying ACCESS:, and pressing **Return** results in the following listing appearing on the screen:

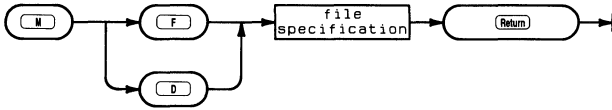
```
ACCESS:          Directory type= LIF level 1
created 20-Sep-82 13,57,17 block size=256
Storage order
...file name...  # blks   # bytes  last chng

FILER           218     55808  20-Sep-82
EDITOR          224     57344  20-Sep-82
LIBRARIAN       202     51712  20-Sep-82
MEDIAINIT.CODE  132     33792  20-Sep-82
TAPEBKUP.CODE   54     13824  20-Sep-82
FILES shown=5 allocated=5 unallocated=3
BLOCKS (256 bytes) used=830 unused=223 largest space=223
```

To get more information about the files and unused areas on the volume, use the Extended Directory command.

Make

The Make command creates files and directories.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	

The Make prompt is:

```
Make File or Directory ? (F/D)
```

The Make command is useful primarily in two ways. Files can be made when you need to reserve physical space on a disc, and directories can be made on an SRM system.

Making Files

The Make command is not required to create files to be used by the various Pascal subsystems. It reserves space only; it in no way initializes or changes the contents of the space. In the Pascal operating system, each subsystem lets you either create or specify any files you need.

The Make command requires at least a file specification (which includes a volume specification by definition) and accepts an optional size specification. If the size (a positive integer) is given, it must follow the file specification on the same line and be enclosed in square brackets. The Filer then creates a file with the specified name and of the specified size on the first area of the volume that has a large enough area of contiguous storage space to meet the size requirements.

When using a size specification to make a file, you must be aware that the size is specified in “number of blocks”. The size of all Make blocks is 512 bytes – regardless of the directory type. A LIF directory considers a 256-byte sector to be a block. The WS1.0 directory considers a block to be 512 bytes. So if you make a file on a LIF volume and specify 500 blocks, it will show up in the directory as 1000 blocks.

For example, assume that there is a volume named MKWORK: on-line that has at least 22 blocks of contiguous and unused space available. Press to initiate Make, specifying:

```
MKWORK:DUX.TEXT[22] 
```

This results in a file named DUX.TEXT being created on the first available area with 22 blocks of the volume MKWORK: and the Filer reporting the following:

```
MKWORK:DUX.TEXT made
```

A subsequent listing of the directory (using the List directory or Extended directory commands) will show a file of the same name with a 22-block size (on WS1.0 directories).

The size specification may be omitted, in which case the Filer creates the specified file using the largest unused area on the disc. (That is, the largest contiguous storage space on the disc will be allocated to the file.) It is recommended that you specify the size you want the file to be.

There are two special cases of size specification worth knowing about. The first is the number zero enclosed in brackets [0], which is the same as omitting the size specification altogether – the Filer uses the largest space available. The second case is the asterisk character enclosed in brackets [*], which tells the Filer to make the file size either the second largest area on the disc or half of the largest area, whichever is greater.

Rebuilding Files

The Make command is useful if you must rebuild a file that was lost on a disc. Here are steps you can take to do that.

1. You must know its size and where it was located.
2. Make TEMP files (TEMP1, TEMP2, etc.) over all the unused spaces on the disc that are as large as or larger than the file you will be making.
3. Make a file of the proper type over the lost file to recover it.
4. Use the Remove command from the Filer to remove all the TEMP files.

An Extended directory listing can help you determine the location and size of unused areas on the disc.

The preceding technique will not recapture lost files on SRM systems.

Making Directories (SRM)

The Make command is used to create directories on an SRM system. For example:

```
Make File or Directory ? (F/D)
```

Answer the first question by typing , and specify where you want the directory located and its name. The directory path tells where you want it and its name is the name on the end of the path. For example, if you had an SRM directory you would type:

```
#5:USERS/JOE/PROJECT1
```

If you wanted to create a directory for Project 1's DATA files, you should type:

```
#5:USERS/JOE/PROJECT1/DATA
```

The DATA directory is created in the PROJECT1 directory.

New

The New command releases or clears the workfile(s).



The New command requires no specifications. Upon pressing to initiate the command, it clears the workfile unless the workfile has been updated since the last Save command. If the workfile has been updated but not Saved, the following prompt appears:

```
Throw away current workfile? (Y/N)
```

Responding by pressing for No allows you to use the Save command to write the file to a volume; for Yes clears the current workfile area.

You can check the status of the workfile before using New with the What command. The What command gives you the name and status (saved or not) of the current workfile.

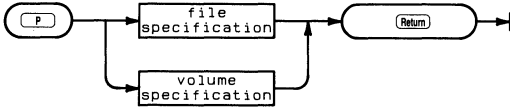
After the Filer executes the New command, it will respond with:

```
Workfile cleared
```

Do not confuse the Filer's New command with the Newsysvol (new system volume) command at the Pascal command line – the two commands are different and perform separate functions.

Prefix

The Prefix command changes the default volume to the one specified. This command performs the same function as the PREFIX command available in the HP EGS Graphics Editor.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	for SRM only
volume specification	complete volume name	

The Prefix prompt is:

```
Prefix to what directory ?
```

The Prefix command is initiated by pressing **P** and it requires a volume specification. The command allows you to specify a new default volume – the volume where the Filer searches for file names when a volume name is not specified. The volume must be block structured (one used for mass storage) but does not have to be on-line. The current prefix (default) volume can be obtained by responding to the Prefix prompt with a colon (:). (The Volumes command may also be used to determine the current prefix volume.)

When the command executes, the screen displays the message:

```
Prefix is MKWORK:
```

MKWORK: is the name of the current default prefix. The Prefix command saves keystrokes if you are doing a lot of file accessing on a particular volume.

Filer commands that request a volume specification may be answered with a colon (:), which specifies the current default volume.

It is possible to set the default prefix to a flexible disc drive, regardless of the volume inside. This is done by typing the following while the drive door is open:

#3: **Return**

The prefix command is used to set up a working directory on a Shared Resource Management system as well. If you had an SRM file named

#5:USERS/JOE/PROJECT1/PROGRAMS/FILE

you could initiate the Prefix command as usual and specify

#5:USERS/JOE/PROJECT1/PROGRAMS

This sets the SRM volume as the default volume (with volume name of PROGRAMS) and USERS/JOE/PROJECT1/PROGRAMS as the working directory on the SRM. Now you can specify the file with:

FILE

If you were to use the Prefix command again to set the default prefix to another volume (not on the same unit), the working directory and volume name for the unit remain PROGRAMS. You need only specify:

#5:FILE

or

PROGRAMS:FILE

Either will get the same file.

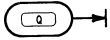
Note

Do not use the Prefix command on unit #45. This is the system volume and should not be altered.

It is possible to change the working directory on an SRM unit without changing the default volume. Use the Unit directory command from the Filer. Press **U** and then give the directory name that you wish to become the working directory. If the new directory is in the existing working directory, just type the new directory name. If it is not, type the whole directory path as shown above in the Prefix example.

Quit

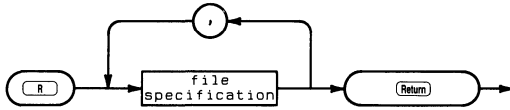
The Quit command exits the Filer subsystem and returns you either to the main HP EGS Manager menu or to the Pascal command line.



The Quit command has no parameters, and no specifications of any type are needed. Press **Q** to exit from the Filer and redisplay the HP EGS Manager menu or Pascal command line.

Remove

The Remove command purges specified files from the directory.



Item	Description/Default	Range Restrictions
file specification	complete volume and file specification	

The Remove prompt is:

```
Remove what file ?
```

The Remove command is initiated by pressing and it requires a file specification. The command removes the specified file from the directory, updates the directory, and reports the action it has performed. Wildcards may be used to specify a subset of files to be removed. If the question mark wildcard (?) is used, the Filer asks you if you want to remove each file in the volume on a case-by-case basis. If the equals wildcard (=) is used in the file specification, the Filer reports the specified file or files and then prompts:

```
Proceed with remove ? (Y/N)
```

This is your last chance to change your mind about the removal. Pressing for No aborts the operation and no files are removed. Pressing for Yes removes those files meeting the wildcard specification from the directory. The process is not always reversible. However, the Make command can sometimes be used to recover a removed file.

Note

The Filer considers the file specification = to specify all the files on the default volume and MKWORK: = to specify all the files on the MKWORK: volume. If you use the wildcard in this form and respond to the Filer's prompt (Proceed with remove ? (Y/N)) with for Yes, every file on the directory of the specified volume is removed. Responding with for No aborts the operation. Wildcards can be hazardous to your files – watch the prompts.

Specifying a single file (of an on-line volume) in response to the Remove prompt results in the removal of that file from the directory and a report that the file has been removed. Once the key is pressed following the file specification (unless wildcards are used), that file is gone.

While the use of the = wildcard results in being prompted for whether you want the directory updated, the ? wildcard acts slightly differently. It allows you to be more selective in your removal. Given the volume PROCESS: containing the files:

```
NOVMEMO.TEXT
MARKLTR.TEXT
PARSER.TEXT
PARSER.CODE
GARBAGE.TEXT
```

When you specify PROCESS:?TEXT in response to the Remove prompt, the screen clears and the following message appears.

```
Remove NOVMEMO.TEXT ? (Y/N)
```

Answering with either or results in this next prompt appearing below the first:

```
Remove MARKLTR.TEXT ? (Y/N)
```

The process continues until you have been prompted for all the TEXT files on the PROCESS: volume. Then the final prompt appears:

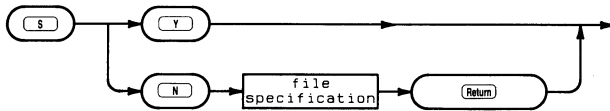
```
Proceed with remove ? (Y/N)
```

You may respond with either for Yes or for No. The files are not actually removed until this prompt is answered with . The ? wildcard thus allows you to be both selective and relatively safe about your file removals.

The Remove operation treats SRM directories like files if they are empty. Remove is not allowed on non-empty SRM directories.

Save

The Save command saves the current workfile on the specified volume.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	

The Save command is initiated by pressing and may or may not require a file specification.

If the workfile was previously named using the Save command, or originally obtained using the Get command, then the Filer prompts:

```
Save as PREVIOUS.TEXT ? (Y/N)
```

PREVIOUS.TEXT is the name previously associated with the workfile. Responding with for Yes results in either a CODE or TEXT file (or both, depending on what is in the workfile) of that name being removed and replaced with the current workfile.

If the workfile was never updated, it is automatically saved with the original name.

If the workfile is not named, or if you answer , the Filer prompts:

```
Save as what file ?
```

When naming the file, the following conventions apply to the type of the file:

1. If a standard suffix is recognized, the workfile is either Filecopied, Translated, or Changed (on the system volume) to the file name and type.
2. If no suffix is recognized, a .TEXT file is the default.
3. If no suffix is included, but a trailing period (.) is found, the file type is Data.
4. The .CODE file is created by removing the suffix (if there is one) and adding .CODE to the file name.

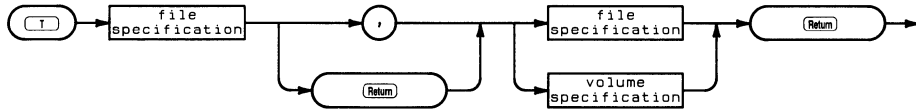
156 Filer Reference

The Filer displays that the file is now saved.

To find out the current name and state (saved or not) of the workfile, use the What command from the Filer.

Translate

The Translate command converts text files between the TEXT, ASCII, and Data types.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	
volume specification	complete volume name	

The Translate prompt is:

```
Translate what file ?
```

The Translate command is initiated by pressing **T** and requires two specifications – a source and a destination separated by either a comma (,) or **Return**. The source specification can be any block-structured volume, any file, or any group of files on a volume. The destination specified can be any of the above and may also be a non-block-structured volume (the **PRINTER:** or **CONSOLE:**). Non-block-structured volumes (like the **PRINTER:**) are assumed to be on-line.

Wildcards may be used to specify sets of files but if a wildcard is used in the source specification, either a wildcard or the \$ character must be included for the destination. If the = wildcard is used, the Translate is not confirmed before taking place. Also, note that if the = wildcard is used alone (without any qualifying strings such as ,TEXT or ,CODE) then the Filer Translates every file on the specified volume. If the ? wildcard is used, you are asked to verify the translate of each file meeting the wildcard specification before the Translate takes place. Thus, using the ? wildcard allows you more flexibility and control over the process.

The dollar sign wildcard (\$) may be used in the destination specification to indicate that the file(s) will have the same name (or names) as the source file(s). For example, assuming that there are a number of TEXT files on the volume DRAW: and that a second volume named MINE: exists,

```
DRAW:=TEXT,MINE:$
```

This results in all the files on the DRAW: volume whose file names end with the string TEXT being translated to the volume MINE: and given the same name as they have on the DRAW: volume.

On a single-volume system, the Translate command proceeds by reading the specified file or files into memory, prompting you to remove that volume and insert the destination volume, and then writing the file(s) in memory to the destination volume. Depending on the amount of memory in your computer and the amount of material being translated, you may have to swap discs more than once.

Note

When using the Translate command with a single-volume system, wait for the Filer's prompt before removing the source volume and replacing it with the destination volume. Failure to follow this guideline may result in the loss of information from the source volume.

The Translate command allows the translating of files or groups of files to non-block-structured devices like the PRINTER: and CONSOLE: . Only text files (of type TEXT, ASCII, or Data) should be sent to printers because other files are not generally human-readable.

When the Translate command has finished its task, the screen displays what file(s) have been translated and the Filer prompt appears. You can abort the Translate command before all specifications are given by pressing **Return** .

In cases where the destination volume already contains a file with the same name as the file being Translated, this prompt is displayed:

```
ANYVOL:XFILE
exists ., Remove, Overwrite, Neither ? (R/O/N)
```

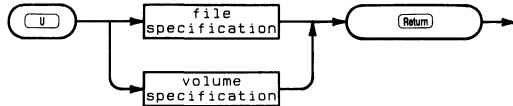
You have the options:

- **Remove:** Remove the existing file before proceeding with the translation.
- **Overwrite:** Replace the contents of the old file with the new information. The Overwrite option cannot be used to change the type of a file on SRM. Attempting to do so will result in the file contents being inconsistent with the file type.
- **Neither:** Cancel the operation.

The Overwrite option allows you to put a file in the same starting location as the original. This is important to SRM users when duplicate links, passwords, and so on exist to a file. All links and passwords to the file are accurate when a file is updated because it is put in the same logical location. If you choose the Remove option, the original file is not actually removed; only your link to it is removed. Any other users are still linked to the original file.

Unit directory

The Unit directory command changes the volume name and working directory for an SRM unit.



Item	Description/Default	Range Restrictions
file specification	complete volume and file name	

The Unit directory prompt is:

```
Set unit to what directory ?
```

The Unit command changes the working directory on SRM units. The working directory and the volume name for SRM units are the same. The Prefix command performs the same operation but sets the default volume to the SRM volume. The Unit command does not.

To specify the working directory, you must start either from the existing working directory or from the root directory.

To get to the root, a unit assigned to an SRM directory must be given or the default is assumed. Follow the unit number with a slash mark (e.g., #5:/). This positions you in the root directory.

From the working directory, you can continue down the tree structure from directory to directory, or you can go back up the structure one directory at a time using “..” for the parent of the current directory.

160 Filer Reference

For example, if the present working directory for unit #5 is:

```
USERS/JOE/PROJECT1/PROGRAMS
```

and you wanted the new working directory to be:

```
USERS/JOE/PROJECT5/DOCUMENTS
```

you can specify it in one of the following ways.

```
#5:/USERS/JOE/PROJECT5/DOCUMENTS
```

or:

```
PROGRAMS:/USERS/JOE/PROJECT5/DOCUMENTS
```

or:

```
#5:../../../../PROJECT5/DOCUMENTS
```

or:

```
PROGRAMS:../../../../PROJECT5/DOCUMENTS
```


Volumes

The Volumes command lists the volumes currently on-line. This command performs the same function as the VOLUMES command available in the HP EGS Graphics Editor.



The Volumes command requires no specifications. Pressing displays the following information about all on-line volumes currently associated with the HP EGS File System: the logical unit number associated with a volume; whether the volume is the system (boot) volume, a block-structured volume, or a non-block-structured volume; the volume name; and the current Prefix or default volume.

Following is a typical display generated by the Volumes command:

```
Volumes on-line:
 1  CONSOLE:
 2  SYSTEM:
 3 # MINI3:
 4 # MINI4:
 5 # MY_SRM:
 6  PRINTER:
45 * SYSTEM04:
Prefix is - MY_SRM:
```

The number on the far left is the logical unit number associated with the volume. The * character in the second column indicates the system volume, which is always block structured. The # character indicates all other block-structured volumes currently on-line. The remaining volumes (shown with no character in the second column) are non-block-structured. The last line of the display shows the current default volume, which is where the system looks for a file when no volume has been specified.

The above configuration shows two flexible disc drives associated with units #3 and #4, and two SRM volumes associated with units #5 and #45; the SRM units are the working volume and system volume, respectively.

What

The What command displays the name and state (saved or not) of the workfile.



The What command is initiated by pressing and requires no other input. The command shows the name of the current workfile or indicates that it is not associated with a file name. It also shows whether the workfile has been Saved since the last update to the file. If no workfile exists, the Filer responds with:

```
No workfile
```

Suppose you had two files named `INFRARED.TEXT` and `INFRARED.CODE` on the default or prefix volume. Assume that you used the Filer's Get command and specified `INFRARED` to associate the files with the workfile. If you then edited the `TEXT` version of that file (using the Pascal Editor), returned to the Filer and executed the What command, the screen would display:

```
Workfile is INFRARED (not saved)
```

because the workfile was changed since the last time a Save command was executed.

Saving the workfile does not change the fact that the workfile exists. It is still there. The New command is used to clear the workfile.

Saving the workfile is not remembered between separate sessions of the Filer. If you Save the workfile during the current Filer session, a New command immediately clears the workfile. If you Save it, quit the Filer and then return to use the New command, the Filer will ask:

```
Throw away current workfile ? (Y/N)
```

even though you saved it during the previous Filer session and have not updated it since.

Zero

The Zero command creates an empty directory on the specified volume. (The Zero command is not allowed on SRM volumes. Use the Make command to create an SRM volume.)



Item	Description/Default	Range Restrictions
volume specification	complete volume name	

The Zero prompt is:

```
Zero directory (NOT valid on SRM type units)
Zero what directory ?
```

The Zero command is initiated by pressing **Z** and requires the volume specification of a block-structured volume. The volume must be formatted either by using the HP EGS system utility for initializing media, or by executing a Pascal program named MEDIAINIT.CODE (which is located in #11: on a local disc and in #45: on an SRM).

Because the Zero command creates a new empty directory on the volume, this prompt appears:

```
Destroy THISVOL: ? (Y/N)
```

Responding with **N** for No aborts the command and returns the Filer prompt.

If you answer **Y**, the next prompt is:

```
Number of directory entries (80) ?
```

The number in the parentheses is the number in the existing directory. Respond with **Return** if that is the number you want. If there is no number in parentheses, **Return** causes the default number for that directory type (80 for LIF; 77 for WS1.0) to be put on the disc.

The next prompt is:

```
Number of bytes (270336) ?
```

This prompt is asking for the logical size of the disc (the extent to be managed by the directory). The number in the parentheses is the number in the existing directory or the default for that disc. Press to use the displayed number, or type in a new number.

The next prompt is:

```
New volume name ?
```

The Filer is asking for a legal volume name. Volume name formats vary with different directory structures. LIF directories allow up to 6 characters with uppercase and lowercase characters being distinct. WS1.0 directories allow up to 7 characters, and all are made uppercase.

Pressing aborts the Zero command.

After typing a volume name, the final prompt appears:

```
NEWSTUF: correct ?
```

Responding with aborts the Zero command. Responding with results in the message:

```
NEWSTUF: zeroed
```

NEWSTUF: is the name of the new volume. The Filer prompt reappears when the operation is complete.

Note

Because the Pascal File System works with volume names, a LIF volume whose name is all blanks (ASCII spaces) will not be recognized as a valid volume.

Appendices

Appendix A: Flexible Disc Handling Guidelines

Inserting a Disc Into Your Disc Drive	167
Inserting a Micro Disc	167
Inserting a Manual Shutter Disc	168
Removing a Micro Disc	168
Inserting a Mini Disc	169
Removing a Mini Disc	169

Appendix B: Run-Time Errors

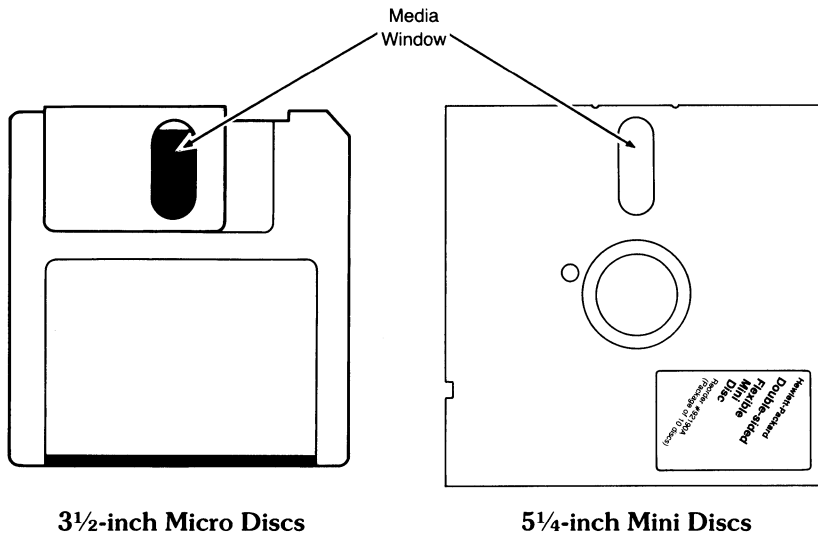
Flexible Disc Handling Guidelines

Appendix

A

Your discs represent a considerable software investment, so treat them with respect. Careless handling can dramatically reduce the life of a disc.

Before reading the guidelines, determine which type of disc you're using: either the 3 1/2-inch micro flexible discs, or the 5 2/4-inch mini flexible discs. Look on the disc label, and use the following illustrations to help you.



The guidelines are:

Back up Discs Frequently

There is a chance of losing information stored on a disc anytime you are using your computer. There are many possible causes for this: a programming bug, operator error, power failure, hardware failure, or disc media failure from wear, damage or contamination. The only sure protection against information loss is to back up your discs frequently.

Cover the Media Window When Not in Use

This is the single most important thing to remember about handling your disc. Micro discs have a metal slide (disc guard) that covers the media window; mini discs have a protective storage envelope. Cover the window to protect the media against dust, finger prints and scratches.

Operate Your Computer in a Clean Environment

Keep your discs away from contaminants like chemical vapors, dust, smoke, ashes, eraser crumbs, and food particles. Never attempt to blow small particles from the disc; the disc cleans itself as it rotates in the disc drive.

Maintain Proper Temperature and Humidity

The proper operating range is 10° C (45° F) to 45° C (115° F) and 20% to 80% relative humidity. Although the disc will operate outside these normal ranges, it will wear out faster and be more prone to errors.

Avoid Magnetic Fields

Data is stored on the disc magnetically, and can be erased by an external magnetic field. Avoid placing the disc near power transformers, magnets, large disc memories or motors.

Remove the Disc from the Drive When Not in Use

Remove the disc completely from the drive when you are through using it. Store it upright in a dust-free container.

Use a Felt Tip Pen to Label Your Disc

Use a soft felt tip pen to label your disc, and be careful to write only in the label area. Using a hard tip pen, such as a ball point, can damage the media.

Don't Touch the Surface of the Disc

Be careful not to touch the media surface through the media window. The thickness of a fingerprint is enough to cause errors. The oil in a fingerprint will also collect dust and cause a disc to wear out prematurely.

Don't Bend or Fold the Disc

The disc is flexible but will not operate if creased. Using ball point pens, rubberbands, paper clips, etc. can crease the disc.

Don't Attempt to Clean the Disc

The inside surface of the disc jacket is covered with a special material that cleans the disc as it rotates in the drive. Any other method of cleaning – including blowing dust from the surface – may damage the media and cause information loss. If a disc becomes dirty or scratched, immediately transfer the data to a new disc and dispose of the old one.

CAUTION

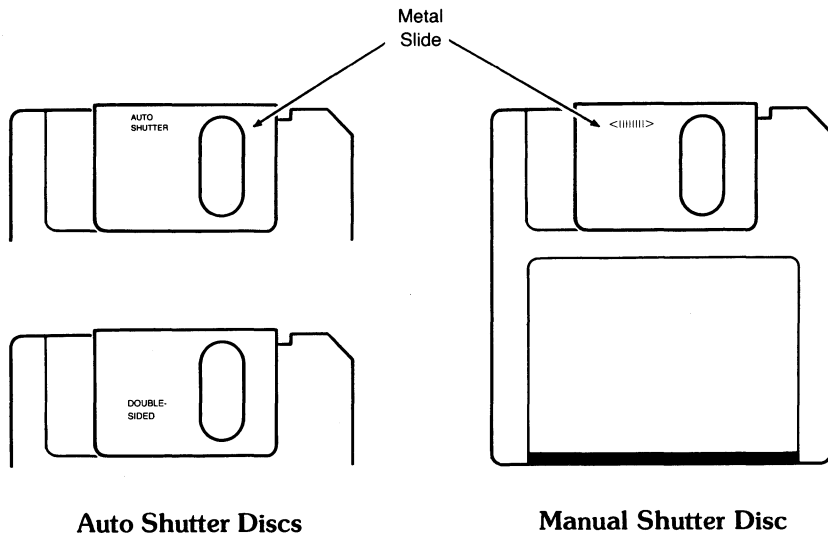
IF YOUR DISC DRIVE EVER DESTROYS THE MEDIA ON A DISC, STOP USING THE DRIVE UNTIL IT CAN BE SERVICED. THIS IS EXCEPTIONALLY IMPORTANT, AS CONTINUED USE OF THE DRIVE WILL DESTROY MORE MEDIA. IMMEDIATELY CALL YOUR NEAREST HEWLETT-PACKARD SALES AND SERVICE OFFICE.

Inserting a Disc Into Your Disc Drive

Inserting a disc into a disc drive is no obvious matter if you have never done it before. The following paragraphs provide instructions for both mini and micro discs; read only the section that pertains to your disc size.

Inserting a Micro Disc

Micro discs come in two varieties: “auto shutter” and “manual shutter.” To find out which type you have, look at the metal slide. If the words “AUTO SHUTTER” or “DOUBLE-SIDED” are printed on the metal slide, you have an auto shutter disc. If a two-directional arrow appears, you have a manual shutter disc. See the following illustrations.



Note

When possible, turn the disc drive on before inserting a disc.

Inserting an Auto Shutter Disc

Slide the disc into the disc drive, label side up and metal slide facing the disc drive.

The drive should completely “swallow” the disc and flash its light to signal acceptance. If the drive rejects the disc and pushes it back out at you, follow these steps:

1. Remove the disc and make sure you inserted it label side up. Also check that the disc drive is turned on. If you inserted the disc incorrectly, try it again.
2. If you inserted the disc correctly but the drive still refuses it, move the metal slide to the left, exposing the media. Re-insert the disc.

Inserting a Manual Shutter Disc

To insert manual shutter micro discs into your disc drive, follow this procedure:

1. Move the metal slide all the way to the left, exposing the media.
2. Insert the disc into the drive, label-side up and metal end facing the disc drive.

The drive should completely “swallow” the disc and flash its light to signal acceptance. If it pushes the disc back out at you, check that the label is facing up and the drive is turned on. Re-insert the disc.

Removing a Micro Disc

Make sure the disc drive is turned on and press the disc eject button in the lower-right corner of the drive. Remove the disc and close the metal slide over the media window if it is not already covered. To close auto shutter discs, pinch the corner to the left of the metal slide (marked “PINCH”).

Inserting a Mini Disc

Note

When possible, turn the disc drive on before inserting a disc.

To insert a mini disc into your disc drive, follow this procedure:

1. Open the drive door by lifting the door handle. Check that there is not already a disc in the drive.
2. Insert the disc into the drive, label-side up with the media window facing the drive.
3. Close the drive door. If the door will not close, push the disc farther into the drive and try again.

CAUTION

IF YOU ACCIDENTALLY INSERT ANOTHER DISC WHEN ONE IS ALREADY IN THE DRIVE, REMOVE THE BOTTOM DISC FIRST. OTHERWISE, THE DISC DRIVE COULD BE DAMAGED.

Removing a Mini Disc

To remove a mini disc from your disc drive, follow this procedure:

1. Make sure the disc drive is turned on.
2. Lift the drive door and carefully pull the disc out.
3. Return the disc to its protective envelope and store it upright in a dust free box.
4. Close the drive door.

Run-Time Errors

Appendix

B

HP EGS System Errors are typically operating system run-time errors. While theoretically any run-time error can occur, only six errors commonly occur:

-2 Not Enough Memory. This happens when you don't have enough RAM to run a program or utility (such as connection lists).

-4 Integer Overflow. The math operation results in an integer value which is out of range.

-5 Divide by Zero. An operation on the data causes this error.

-10 Non-zero IORESULT. This happens when an I/O operation failed. Causes of this are hardware failure, a peripheral not being on line, or media not being found.

-12 CPU Bus Error. Range checking is turned off by HP EGS. This can result in the workstation incorrectly reporting an error.

-27 Graphics routine error. This typically happens when something goes wrong with the plotter hardware or the interface to the plotter.

If the problem persists, try to re-create the error using an echo file to log the problem.

172 Run-Time Errors

HP EGS system errors:

- 0 Normal termination...No error
- 1 Abnormal termination
- 2 Not enough memory...Insufficient RAM available to run the program
- 3 Reference to NIL pointer
- 4 Integer overflow
- 5 Divide by zero
- 6 Real math overflow (The number was too large.)
- 7 Real math underflow
- 8 Value range error
- 9 Case value range error
- 10 Non-zero IORESULT
- 11 CPU word access to odd address
- 13 Illegal CPU instruction
- 14 CPU privilege violation
- 15 Bad argument - SIN/COS
- 16 Bad argument - Natural Log
- 17 Bad argument - SQRT
- 18 Bad argument - real/BCD Conversion
- 19 Bad argument - BCD/real/ Conversion
- 20 Stopped by user
- 21 Unassigned CPU trap
- 22 Reserved
- 23 Reserved
- 24 G80x0 macro parameter out of range
- 25 Undefined G80x0 macro parameter
- 26 Error in I/O subsystem
- 27 Graphics routine error
- 28 Parity error in memory
- 29 Misc. floating-point hardware error

Subject Index

a

Aborting commands while in the text editor	M 31
Access	M 128
Activating tablet menus	L 285, S 197, U 90
Active:	
Commands	S 5
Drawing area	U 86
ACTUAL:	
DRILL (Photoplot/Drill)	U 207
PHOTOPLOT (Photoplot/Drill)	U 207
AdBox	L 55
AdCom.	L 55
ADD	S 7
ADD (archive)	S 244-245
ADD ARC	S 10
ADD ARC (archive)	S 246
ADD CIRCLE	S 14
ADD CIRCLE (archive)	S 247
ADD DIMENSION	S 18
ADD DIMENSION (archive)	S 248-249
ADD HATCH	S 24-28
ADD HATCH (archive)	S 250-251
ADD instance	S 30-31
ADD INSTANCE (archive)	S 252-253
ADD LINE	S 32-37
ADD LINE (archive)	S 254
ADD MARKER	S 39
ADD MARKER (archive)	S 255
ADD NOTE	S 48-52
ADD NOTE (archive)	S 260-261

ADD OVAL	S 40-42
ADD OVAL (archive)	S 256-257
ADD POLYGON	S 44-45
ADD POLYGON (archive)	S 258
ADD RECTANGLE	S 46-47
ADD RECTANGLE (archive)	S 259
ADD TEXT	S 48-52
ADD TEXT (archive)	S 260-261
ADD*	L 23
Adding:	
Angular dimension	L 122
Arcs	L 40, L 107, S 10
Bevel	S 34
Beveled face	L 86
Board blank outline	L 250
Bolt	L 134
Bolt head	L 144
Capacitors and resistors	L 180, L 236
Center lines	L 108, L 109, L 118
Circle	L 30, S 14
Connections	L 192
Construction circles	L 106
Construction lines	L 59, L 105, L 112, S 33
Diameter holes	L 108
Diametral dimensions	L 123
Dimension notes	L 124
Edge connector	L 249
Flange	L 120
Flange part	L 132, L 148
Front view dimensions	L 125
Hole	L 88
Isometric rectangle	L 84
Keepout area	L 240
Leader lines	L 151, S 33
Linear dimensions	L 167
Logical ports	L 187
Material list	L 162
Multivertex lines	L 43
Note	L 94, S 43-52
Nut and washer items	L 149
Outline	L 195
Pad set	L 238
Part type	L 186, L 243

Physical ports	L 188, L 244
Polygon	L 32, S 44-45
Primitive	L 23
Radial dimensions	L 123
Reference designator	L 186, L 243
Roof line	L 28
Semicircle and tangent lines	L 106
Solder resist	L 227
Source voltage input	L 194
Surface finish symbol	L 128
Text while in the text editor	M 29, M 102
Tooling marks	L 262
Traces	L 261
Vertex	L 60
Vertical dimensions	L 126
Washer and nut	L 136
Adjust	M 93
Adjusting margins	M 45, M 93
Aligning a part	L 185, L 242
Alignment of grid with GRID command	S 136
*ALL	L 44, L 102
Alpha:	
Dump	S 103
Screen	L 16, U 7
Analog parts	L 279
*ANG	L 28, L 93
*ANGL	L 41
Angle, lock angle warning	S 160-161
*ANGS	L 40
Angular dimension	L 122, S 19
Annotating:	
Block	L 90
Flange drawing	L 120
Annotatn mode	L 101
Apertures	U 181
Application notes	U 259
Applications, user-written	U 46
ARC, ADD	S 10
Arcs, adding	L 40, L 107
Arc methods	L 40
ARCHIVE	L 77, S 53-55, U 239

Archive file:	
Commands used in	S 244-296
Description	S 236-242, U 24-27
\$FILES	S 239
Process file in	S 236
Retrieving drawings and instances with EDIT	S 239
Retrieving macro instances with DEF :I	S 240
Using to transport data	U 259
Archive Reference	S 235-296
Archived drawings, retrieving with RETRIEVE	S 192
Archiving drawings	U 137
AREA	S 56-57
AREA, isometric	S 57
AREA?	L 53
Array:	
Creating with STEP	S 204-207
Step specification	S 8
ASCII ordinals	U 98
Assembler	M 73
Assembling parts	L 132
Assembly drawing:	
Annotating	L 147
Creating	L 268
Material listing	L 156
Associated text:	
:AC text	S 50, U 20-21, U 136
:AD text	S 50, U 20-21, U 136
Description	U 20-21
Drawing	U 78, U 87
Tag values	U 136, U 197
Attribute data	U 171, U 203-206
ATTRIBUTES (Photoplot/Drill)	U 203
AUTOFLASH (Photoplot/Drill)	U 187, U 206

b

:B, used to defined a hatched region	S 25
BACK SPACE	L 5, L 6
Backup (^)	S 58
Bad sector.	M 130
*BASE	L 167
Base line, dimension.	S 19-20
Basic layers	L 227
BEGIN_ASSOC	S 262
Bevel, ADD LINE	S 34
Beveled face, adding.	L 86
Bill of materials, creating.	L 212
BK, backup for HP EGS personalities.	L 26
*BLNK.	L 103, L 147
Block letters (text)	S 48-52
Board blank outline, adding.	L 250
Bolt head, adding	L 144
Bolt, adding	L 134
Boot-up display	I 22
BOTH.	L 44
Breaking up large files	M 15
BUBBLE	L 103
Building:	
Control file	U 42
Macro file	U 62
Process file	U 52
Screen menu	U 57
Start files.	U 48
Tablet menu.	U 85
Volume search tables	U 50
Bus addresses, setting on a peripheral	I 16-17
Bus notation (Connection Lister)	U 129
Busline	L 177
Buslines for connections.	L 216
Bytes, seeing the number of at boot-up.	I 22

C

*C&C	L 30
*C&Rs	L 102
Calculating expressions with EVALUATE	S 116
CALIBRATE	S 59-61
Capacitors and resistors, adding	L 180, L 236
CATALOG	S 62-63
Center lines and fillets, deleting	L 139
Center lines, adding	L 108, L 109, L 118
CENTR	L 38
Certificate, right-to-use (codeword)	I 43, U 175, U 233
Chain copying	S 67-70
*CHAM	L 28
Chamfer:	
ADD LINE	S 34
Dimensioning	L 127
CHANGE	L 80, S 64-65
Change	M 60, M 131
Changing:	
Color	L 51
Default volume	M 54, M 150
File names	M 60, M 131
File types	M 61, M 153
Parts file	L 156
Tolerancing option	L 93
Width	L 51
Character cell	U 97
Character code, converting	M 16
CIRCLE:	
ADD	S 14
Methods	L 30
CLR -> END	L 5
CLR I/O	L 5
CLR LN	L 5, L 6
*CMPT	L 103
*CNTR	L 102

Codeword:	
Installation utility	I 44-46
Installing one codeword	I 46
Installing several codewords at once	I 47-48
Obtaining	I 44-45
Right-to-use certificate	I 43, U 175, U 233
Collecting part information	L 147
Collections, building, modifying, or destroying with GATHER.	S 123-128
COLOR (SHOW* command)	L 44
Color:	
Changing	L 51
Cursor	S 73-75
Defining component colors with EQUATE	S 115
Grid	S 135
Of each entry	U 58
Screen menu	S 164
Command entry:	
Editing keys	U 9-10
Files	U 3
Graphics input device	U 3
Keyboard	U 3
Screen menu	U 3
Tablet menu	U 3
Command types:	
Commands to define other commands and macros.	U 32-33
Current	S 5, U 11-12
Default	U 32-33
Drawing creation and editing	U 32-33
File and volume manipulation	U 32-33
Information and drawing environment	U 32-33
Input and output	U 32-33
Primary	S 3, U 10-11
Secondary	S 3, U 10-12

Commands:

Definition	U 2
Entering.	L 12, S 4
HP EGS personality.....	S 6
Loading with LOAD	S 157
Methods	L 15
Options	L 15
Primary	S 3
Secondary.....	S 3
Syntax.....	S 1
Terminating	L 25, S 5, S 174
Terminators.....	L 15
Understanding	S 3
Used on screen menus.....	L 15
Communicating with HP EGS	L 12
*COMP	L 141
Comparing connection lists.....	L 264, L 268, U 169
Compiler	M 73
Completing commands.....	S 5
Component:	
Adding with lock angle enforcement	S 161
Constants	U 70
Copying.....	L 25
Copying restricted	L 45
Defining color with EQUATE	S 113
Definition	U 2
Deleting.....	L 31, S 86-87
Descriptors.....	S 2, U 18
Dynamic tracking.....	U 16
Filling.....	L 24
Hatching	S 26-28
Identifying	L 49
Moving	L 70
Moving with MOVE.....	S 172-173
On screen menus	L 15
Restrictors	L 15
Supported by PHOTOPLOT	U 185

Computers used with HP EGS, models 217, 220, 236, 237, 310, and 320	I 8
CONCNST (Connection Lister default string constants)	U 160
Conditional macros	U 75
CONERR (Connection Lister errors)	U 156-159
Configured disc, connecting	I 14-15
Connect	L 177
Connected part, moving	L 193
Connecting:	
Passive parts	L 182
Traces	L 257, L 273
Connection list:	
Comparing	L 264, L 268, U 169
Creating	L 199
Efficient processing	U 147
Format	U 148-156
Making	L 198, L 264
Merging	U 161-164
Preparing	L 198
Utilities	U 161
Connection Lister:	
Archiving drawings	U 137
Associated text	U 136
Codeword	U 132-135, U 146-147
Customizing	U 159-160
Default files	U 142-146
Error file	U 157-159
Installation	U 131-135
Menu options	U 141
Message File	U 146-147, U 159-160
Operational errors	U 156
Output file	U 148-156
Using	U 136-160

Connection listing:	
Bus Notation	U 129
Concepts	U 126-130
Definition of Terms	U 126
Flat	U 128
Hierarchical	U 128
Implicitly stepped components	U 130
Logical rules	U 126
Physical rules	U 126
Properties	U 130
Special reference designators	U 129
Connection hierarchy	U 127
Connections, adding	L 192
*CONS	L 26
Consolidating free space on volume	M 7
Construction circles, adding	L 106
Construction lines:	
Adding	L 26, L 59, L 105, L 112
Deleting	L 115
CONTINUE ON ERROR (Photoplot/Drill)	U 207
Control file:	
Building	U 42
Structure	U 42
Convert:	
Old parts file	U 107
Character code	M 16
Coordinates, locating with POINT	S 187
Copy	M 40, M 95

COPY*	L 25
COPY:	
Chain	S 67-70
Command description	S 66-70
Datum	S 67-70
Copying:	
Components	L 25
Files	M 59, M 138
Files with FILECOPY	S 120-121
Instances	L 68
Restricted components	L 45
Set	L 138
Text in text editor	M 40, M 95
View	L 133
Correcting:	
Pad locations	L 240
Location errors, x,y	L 26
CPU ID Prom	I 45
Cpy	M 40, M 95
Creating:	
Artwork	L 264
Assembly drawing	L 270
Bill of materials	L 212
Circuit side physical layer	L 271
Connection list	L 199
Drawing with EDIT	S 106-111
Ground physical layer	L 272

Holes	L 116
Library parts	U 77-84
Macros with DEFINE	S 80-85
Material list	L 209
Material list data	L 156
Parts list	L 159
Schematic library part	L 183
CRT modules for the INITLIB file	I 68
CTRL BACK SPACE	L 6
CTRL CLR LN.	L 6
Current:	
Command	U 59
Working directory	M 11
CURSOR	S 72-75, U 16, U 22
Cursor:	
In text editor	M 27, M 32
Moving	L 13
Size	S 5, S 72-74
Snapping modes	L 17
User-defined	S 72-75
Customizing:	
Steps	U 35

d

*DATA.....	L 103, L 147
Date:	
Entering.....	L 8
Setting system date.....	M 85
DATUM:	
Command description.....	S 77
Used like COPY :D.....	S 77
Datum:	
Copying.....	S 67-70
Dimensioning.....	S 19
Debugger.....	M 73
DECIMALS.....	S 78
DECIMALS (archive).....	S 263
Default files:	
Connection Lister.....	U 142-146
IGES Translator.....	U 240-244, U 251-252
Photoplot/Drill.....	U 201-203
Default:	
Commands.....	U 32-33
String constants.....	U 160
Values, process file.....	U 53-54
Volume.....	M 6
Volume, specifying with PREFIX.....	S 188
DEFINE.....	S 80-85, U 62, U 79
DEFINE (archive).....	S 264-265
Defining:	
Layers.....	U 52
Macros with DEFINE.....	S 80-85, U 62
Personality.....	U 38
Part geometry.....	L 183
PC board library part.....	L 238

DEL CHR.	L 5
DELETE.	S 86-87
Delete.	M 34, M 97
DELETE*	L 31
Deleting:	
Center lines.	L 145
Center lines and fillets	L 139
Components	L 31
Construction lines.	L 115
Hidden lines and fillets	L 139
Hidden portions	L 137
Text in text editor.	M 34, M 97
Descriptors, component	S 2
Detail layer:	
Defining with EQUATE.	S 113
Type	U 19
Detail mode, setting display with WINDOW	S 227
Detailing the hub.	L 114
*DIAM	L 30
Diameter dimension	S 19
Diameter holes, adding.	L 108
Diametral dimensions, adding	L 123
Differences between HP EGS 2.0 and 2.1.	I 65-66
Digitizer, Summagraphics	I 20-21
Digitizing:	
Drawings with CALIBRATE	S 59-61
Tablet menus with SAVE	S 197
Dimension:	
Extension lines, distance between	S 90-91
Tolerance values	S 18-22
Tolerance values, font size of	S 96
Dimensioning:	
Chamfer	L 127
Circle location.	L 90
Diameter of a circle	L 91
Face of a block	L 91
Fillet radius	L 126

Dimension notes, adding	L 124, S 18-22
Dimension values:	
Font size	S 92
Number of decimal places	S 88
Orientation	S 89
Radix	S 93
Scale	S 95
Slant	S 97
Units	S 98-99
DIMENSION, ADD	S 18
*DIMN	L 103
DIM_DEC	S 88
DIM_DEC (archive)	S 266
DIM_DIR	S 89
DIM_DIR (archive)	S 267
DIM_EXTEND	S 90-91
DIM_EXTEND (archive)	S 268
DIM_FSIZE	S 92
DIM_FSIZE (archive)	S 269
DIM_RADIX	S 93
DIM_RADIX (archive)	S 270
DIM_SCALE	S 95
DIM_SCALE (archive)	S 271
DIM_TFSIZE	S 96
DIM_TFSIZE (archive)	S 272
DIM_TSLANT	S 97
DIM_TSLANT (archive)	S 273
DIM_UNITS	S 98-99
DIM_UNITS (archive)	S 274-275
Directories, listing	M 55, M 144
Disc drive, finding the drive number	I 5
DISPDISP	L 233
DISPLAY	S 100-101
Display:	
Grid, setting with GRID	S 135-137, U 14-15
Optimization layers	L 231
Option switches, setting on a video interface card	I 19
Setting the default with the Codeword utility	I 46
DISPPEN	L 233
DISTANCE	S 102
DIBox	L 55

DICom	L 55
Dlete	M 34, M 97
Drawing:	
Area, changing, zooming, and magnifying with WINDOW	S 227-228
Components	U 18
Creation and editing commands	U 32-33
Environment	U 52
Environments, defining with EQUATE	S 112-115
Files	U 17
Hierarchy	U 30
Hub	L 110
Initializing	L 120
Isometric rectangle	L 82
Location, setting with WINDOW	S 227
Loading with LOAD	S 157
Magnification, setting with WINDOW	S 227
Modifying with MODIFY	S 168-171
Origin placement	U 77
Plotting	L 38
Printing	L 38
Re-editing	L 248
Saving	L 36, L 45
Saving with SAVE	S 196-198
Scale	U 77
Tablet menu	U 87
*DRID	L 154
DRILL:	
Description	U 171
Output	U 194-196
Overview	U 194
Virtual DRILL output	U 195-196
DRILL FORMAT (Photoplot/Drill)	U 208
DRILL ORIGIN (Photoplot/Drill)	U 194, U 210
Drill post-processor:	
Codeword	U 174-178
Installing	U 172-178
Overview	U 194
Using	U 179-180
DRILL UNITS FACTOR (Photoplot/Drill)	U 194, U 213

*DSRY	L 55
DUMP	S 103
DUMP*	L 38
Duplicate link	M 133
Dynamic component tracking	U 16
Dynamic tracking, terminating with NOP	S 174

e

ECHO	S 105
Echo file	S 105
Edge connector, adding	L 249
Edge connectors	L 277, L 282
EDIT	S 106-111
EDIT (archive)	S 276
Editing:	
Keys	U 9-10
Message files	U 94
Editor	M 92
Editor, using to define macros	S 82
Edit_PCB	L 232
Edit_Sch	L 176
EGS/IGES Translator:	
Description	U 239-249
Limitations	U 248-249
Messages	U 247
Relationships	U 238
EGS/IGES Translator default file:	
Default dimension arrowheads	U 243
Dimension arrow	U 242
Form parameters	U 241-242
IGES note fonts	U 242
Leader arrow	U 242
Memo parameters	U 241
Note and text form numbers	U 242
Note and text stroke files	U 241
Tape driver parameters	U 243

Electrical Engineering:

Drawings	L 1
Loading	L 179
Personality files	M 22
Personality, installation on a hard disc	I 30-31
Personality, installation on an SRM	I 41-42
Post-processor	U 103, U 110
Schematic Drawing	L 1, L 173
Electrical schematic connection lister	U 140, U 142
END_ASSOC	S 277
Enforcement of lock angle	S 160-161
Enter	L 23
Entering:	
Commands	L 12
Data	L 5
EOC	L 25
EQUATE	S 112, U 19, U 52
EQUATE (archive)	S 278-28
Errors:	
Connection Lister	U 156-159
IGES Translator	U 247, U 255-256
Photoplot/Drill	U 225-228
Run-time errors	M 171
EVALUATE	S 116-118, U 16
Excellon drill files	U 171, U 194
eXchange	M 36, M 121
eXecute	M 89
Executing a code file	M 89
EXIT	S 119
EXIT (archive)	S 281
EXIT, use of during an instance-in-context EDIT	S 110
Expressions, creating with DEFINE	S 80-85
Extended directory	M 135
External monitor, setting default with codeword utility	I 46

f

f2 key	L 6
f3 key	L 6
File:	
Changing file name	M 60, M 131
Changing file type	M 61, M 153
Concepts	M 1
Copying	M 59, M 138
Copying with FILECOPY	S 120-121
Drawing	U 17
Moving	M 59, M 138
Names	M 2
Operations	L 79
Plotting with PLOT	S 182
Removing with REMOVE	S 190-191
Saving with SAVE	S 196-198
Specifications	M 1
Suffixes	S 65
Table, displaying or changing system file table	M 87
File and volume manipulation commands	U 32-33
File management:	
Breaking up large files	M 15
Re-creating large files	M 15
FILECOPY	L 80, S 120-121
Filecopy	M 59, M 138
Filer	M 126
\$FILES	S 282
FILL	L 24, L 27, L 74
FILL (used with PLOT*)	L 38
Fill modes with PHOTOPLOT	U 186
FILL WIDTH (Photoplot/Drill)	U 208
Fillet	S 11
Fillet radius, dimensioning	L 126
Filling in Items	L 24
Filling paragraphs in text editor	M 46, M 106
*FILT	L 40
Find	M 42, M 99
Flange, adding	L 120

Flange drawing:	
Annotating	L 120
Creating	L 97
Flange part:	
Adding	L 132
Adding items	L 148
Flat connection listing	U 128
Flexible discs:	
Initializing	M 13
Inserting and removing	I 25-26
Use of	I 25-26
*FONT	L 74, L 148
Font size:	
Changing with MODIFY	S 169
Of notes and text	S 49-51
Setting with FSIZE	S 122
FORMAT (Material listing)	U 118-120
Fractional resolution of dimension values	S 99
*FREE	L 28
FREE:	
ADD LINE	S 32-37
ADD POLYGON	S 45
Front view dimensions, adding	L 125
*Front30	L 82
FSIZE	S 122
FSIZE (archive)	S 283
Functions, calculating with EVALUATE	S 117

g

GATHER	S 123-128
GATHER*	L 54
GATHER* options	L 55
gedit.ms.ASC file, entering codeword in	I 47-48
General drawing files	M 19

GENERATE	L 77, S 130-132, U 172, U 239
Generate:	
Files	U 172
Post-processed material list	U 112, U 120
Rat's nest	U 164
Geometry	L 177
Geometry mode	L 101
Gerber photoplotter	U 171
Get	M 141
\$GR	L 17
Graphical EDIT	S 107-108
Graphics and alpha screens	L 16
Graphics:	
Input device	L 13, U 3
Screen	L 16, U 7
Graphics dump	S 103
Graphics editor:	
Description	U 1-2
Files	M 19
Graphics tablet:	
Loading with LOAD	S 156-157
Saving tablet menus with SAVE TMENU	S 197
Setting the bus address	I 16-17
Setting the default with codeword utility	I 46
Toggling on and off with TMENU	S 217
Using to digitize a drawing with CALIBRATE	S 59-61
GRID	S 134-137, U 22
GRID (archive)	S 284
:GRID snapping mode	S 4, U 15
Grid:	
Description of system	U 12-15
Display	S 135-137, U 14-15
System	L 17, S 222-223, U 12-13
User	L 18, U 14-15
Using while building library parts	U 77
GROUP	S 138-139

h

:H, used to define an unhatched region	S 25
Hard disc:	
Copying files	I 57
Emptying	I 58
Finding contents of	I 56-57
Initializing	I 58-59
Listing volume contents	I 57
Removing files	I 58
Setting the bus address	I 17
Hardware:	
For HP EGS	I 7-9
Connecting for Installation	I 8-1
HATCH, ADD	S 24-28
Hatching:	
Changing style with MODIFY	S 169
Options	L 41
With different snapping modes	S 27-28
HB-IB cables, connecting	I 11
HELP	S 141-143
HELP (Photoplot/Drill)	U 209
Help messages	U 96
HELP*	L 34
Hidden lines and fillets, deleting	L 139
Hidden portions, deleting	L 137
Hierarchical:	
Connection listing	U 128
Directory structure	M 8
Hierarchy:	
Connection	U 127
Drawing	U 30
Logical	U 31
*HILO	L 91
*HOLE	L 41
Hole:	
Adding	L 88
Creating	L 116
HORizontal dimension values with DIM_DIR	S 89
Horizontal line dimension	S 19
Hpegs	M 74

HP EGS 2.0, differences between 2.0 and 2.1	I 65-66
HP EGS 2.1:	
Additional enhancements since HP EGS 2.0	I 66
File naming conventions	U 36
Hardware differences since HP EGS 2.0	I 65
Manager	U 50
Option 085, connecting the configured disc	I 14-15
Software installation	I 23-42
Stroke files	U 97
System differences since HP EGS 2.0	I 65
User interface differences since HP EGS 2.0	I 66
HP TechWriter, readying drawings for	L 38
HPTW	L 38
*HrVr	L 90
HrVrt	L 59
Hub:	
Detailing	L 114
Drawing	L 110

i

I-C DIP parts	L 282
ID module, linking to an HP EGS codeword	I 42-46
IDENTIFY	S 145-146
Identifying components	L 49
IFELSE function in macros	U 75-76
IGES Translator:	
Codeword installation	U 232-236
Copious data	U 238
Default File	U 251-252
Directory entry section	U 237
Files	M 25
General note	U 238
Global section	U 237
Limitations	U 256
Local installation	U 231
Messages	U 254-256
Parameter data section	U 237

Retrieving drawings	U 250
SRM installation	U 231
Start section	U 237
Subfigure instance	U 238
Terminate section	U 238
IGNORE identifier	U 129, U 160
Implicitly stepped components	U 130
Implicitly stepped components, smashing with SMASH	S 203
\$IN	L 17
in>mm	L 102
INAME	S 147
INAME (archive)	S 285
*INBT	L 137
Indexing message files	U 94
Information and drawing environment commands	U 32-33
Initialization:	
Explanation of	I 53-55
Performing with the Zero command	I 58-59
Initialize	M 75
Initializing:	
Drawing	L 120, L 132
File system	M 75
Flexible discs	M 13
INITLIB file, modules needed for HP EGS 2.1	I 68
INLine dimension values with DIM_DIR	S 89
Inner layer:	
Pads	L 275, L 281
Planes	L 229
INPUT	L 78, S 148-149
Input and output commands	U 32-33
Input device:	
Graphics	L 13
Described by locator tag	S 159
Input file:	
Creating from a GENERATE file	S 148
Pausing with PAUSE	S 177
INPUT:	
(Photoplot/Drill)	U 209
Call up start file	U 48
In control files	U 46
INS CHR	L 5
Insert	M 29, M 102

Installation:

Connection Lister	U 131-135
EWEE on a hard disc	I 30-31
EWEE on an SRM	I 41-42
EWME on a hard disc	I 30-31
EWME on an SRM	I 41-42
EWSYS on a hard disc	I 29
EWSYS on an SRM	I 40
Hardware needed for	I 7-9
HP EGS codeword	I 42-49
HP EGS on a new hard disc	I 26-31
HP EGS on an SRM	I 37-42
HP EGS on top of Pascal 3.1	I 67-69
HP EGS Option 085	I 23
HP EGS Options 042 and 044	I 23-42
HP EGS Software	I 7
IGES Translator	U 231-236
Pascal 3.1 alongside HP EGS 2.1	I 70
Photoplot/Drill	U 172-178
Stream files	I 27
Terms used	I 5
Verifying	I 50-52
Instance-in-Context EDIT	S 108-111
Instance-oriented connection list	U 148-156
Instance:	
ADD	S 30-31
Adding	L 63
Copying	L 68
Creating with WRAP	S 230-233
Description of	U 28-31, U 126
Editing with EDIT	S 107-108
Saving with SAVE	S 196-198
Smashing	L 69
Used to create library parts	U 78
Viewing	L 182
Interconnect:	
Layer, defining	S 113
Layer type	U 19
Mode, setting display	S 227
Interface card, video	I 18-19

Internal units per user unit, used with UNITS	S 223
Intersect	S 4
:INTERSECT snapping mode	U 15
Isometric:	
Area	S 57
Grid	S 135-137
Grid, cursor in	S 74
Rectangle	L 82, L 84
Isometric drawing:	
Archive files	U 24-28
Converting orthogonal views	U 24-28
Isometric conversions	U 23-24
Procedures	U 21-28
Views (front, top, right)	U 22
ITEM	L 103

j

Jump	M 105
JUNCTION	L 178
*JUST	L 74
Justification value for notes and text	S 49, S 51

k

Keepout area, adding	L 240
Keepout layer, master	L 230
Keyboard cables, connecting	I 12-13
Krunch	M 142

1

Large file management	M 15
Largest space on a disc	S 63
Layer:	
Definitions	L 52, L 99, L 224, U 52
Description	U 18-19
Detail	U 19
Interconnect	U 19
Label	U 52
Number	U 52
Symbolic	U 19
Using while creating library parts	U 77
Layers, defining with EQUATE	S 113
*LEAD	L 26, L 94
Leader line option of ADD LINE	S 33
Leader lines, adding	L 151
LEVEL	S 151, U 52
LEVEL (archive)	S 286
LEVEL*	L 177, L 232
Levels, nesting	L 66
Librarian	M 73
Library parts:	
Creating a schematic	L 183, U 77-84
Description	U 17
Drill holes (Photoplot/Drill)	U 223
Flash apertures (Photoplot/Drill)	U 219-220
Ignore and no fill tags (Photoplot/Drill)	U 222
Macro instances	U 78-82
PC board	L 238, L 278
Polygon, hatch, rectangle fill aperture (Photoplot/Drill)	U 220-222
Schematic	L 218
Special considerations	U 77-78
Text/note draw aperture (Photoplot/Drill)	U 222
LINE	L 28, L 167

Line:

Adding multivertex	L 43
Construction	L 26
Defining type	U 52
Deleting	L 145
Dimension	S 19
Replacing portion with STRETCH	S 214
Stretching	L 142
Style	S 114
Type	S 113
LINE, ADD	S 32-37
Linear dimensions, adding	L 167
LIST	L 78, S 152-154

List:

Directory	M 55, M 144
Parts file	U 109
Post-processed material list	U 122

LIST MATERIAL:

Command	S 153
Output file	U 104, U 105

LIST_MAT	L 103, L 177, L 233
--------------------	---------------------

lkNEW	L 59
-----------------	------

lkON/lkOFF	L 59
----------------------	------

*LMEN	L 260
-----------------	-------

LNTYP	L 44
-----------------	------

*LNUM	L 260
-----------------	-------

LOAD	S 155-158, U 31, U 48, U 52, U 62, U 91, U 96
----------------	---

Loading:

Electrical Engineering	L 179
----------------------------------	-------

General Drawing	L 11
---------------------------	------

Mechanical Drafting	L 104
-------------------------------	-------

Printed Circuit Board Layout	L 235
--	-------

Programs	S 157
--------------------	-------

Tablet Menus	S 157
------------------------	-------

Local mass storage:

Checking readability	M 67, M 130
--------------------------------	-------------

Compacting files	M 67, M 142
----------------------------	-------------

Creating an empty directory	M 68, M 163
---------------------------------------	-------------

Identifying unreliable disc sectors	M 130
---	-------

Location errors, correcting	L 26
---------------------------------------	------

LOCATOR	S 159
-------------------	-------

Locator, setting with the Codeword utility	I 46
--	------

LOCK	S 160-161
LOCK (archive).....	S 287
Lock angle warning	S 160-161
Logical levels.....	L 190, L 240
Logical operators, calculating with EVALUATE.....	S 116-117
Logical ports, adding.....	L 187
Logical:	
Axis	U 21-24
Connection rules	U 127
Level	U 31, U 78
LOG_PORT.....	L 178

m

Macro:

Building or modifying a file.....	U 62
Conditional.....	U 75-76
Creating with DEFINE.....	S 82-85
Description.....	U 62
How to modify existing information.....	U 67
How to save values and set modes	U 67
HP EGS personality.....	S 6
Recursive	U 67
Short vs. long.....	U 63
Substituting for command	U 65
Syntax rules	U 63
Temporary	U 66
Macro constants, system-defined.....	U 68-73

Macro file:

For each personality	L 101, L 176, L 231
Listing file contents with LIST.....	S 153
Loading file with LOAD	S 156
Saving file with SAVE	S 196-198

Macro instance:

Archiving as a graphical instance	S 54
Creating with DEFINE.....	U 77-84, S 82-85
Description.....	U 17, U 77-84
Editing with EDIT	S 108-111, U 77
Library parts	U 17, U 29, U 78-82

Mag/paper tape utility	M 16
*MAKE	L 55
Make	M 146
Make_Prt	L 176, L 232
Making connection lists	L 198, L 264
Making material lists	L 198
Manager menu, explanation of	I 50-51
Manufacture and assembly layers	L 228
Margin	M 46, M 106
MARKER, ADD	S 39
Mass storage, IGES Translator requirements	U 249, U 257
Master keepout layer	L 230
Material list data, creating	L 156
Material list:	
Adding	L 162
Creating	L 198
Description	U 103
Error checking	U 114, U 123
File, preparing with LIST	S 153
Of an assembly drawing	L 156
Output format	U 117
Parts file editor	U 105
Mathematical:	
Calculations with EVALUATE	S 116
Expressions	U 16
Expressions, creating with DEFINE	S 80-85
Matrix, creating radial matrix with STEP	S 204-207
*MDFY	L 55
Mechanical Engineering:	
Drawings	L 1
Personality	L 97
Personality file	M 23
Personality, installation on a hard disc	I 30-31
Personality, installation on an SRM	I 41-42
Personality loading	L 104
Post-processor	U 103, U 115
Tablet menu	L 172
Mechanical library parts	L 168
MEMORY	S 163
Memory:	
Requirements for HP EGS	I 7-9
Used by loading commands	S 157

Memvol.....	M 76
MENU.....	S 164-166, U 57
Menu file, loading with LOAD.....	S 156
Menu:	
Height.....	U 59
Horizontal or vertical.....	U 59
Items.....	L 16
Loading with LOAD.....	S 155-157
Manager.....	I 50-51
Modifying and moving screen menus with MENU.....	S 164-166
Row.....	U 58
Saving tablet menus with SAVE TMENU.....	S 197
Toggling on tablet menus with TMENU.....	S 217
Width.....	U 59
Merging connection lists.....	U 161-164
Message file:	
Customizing (Connection Lister).....	U 159-160
Customizing (Photoplot/Drill).....	U 216-218
Editing and indexing.....	U 94
Loading with LOAD HELP.....	S 156
Modifying.....	U 93-96
Message tags.....	U 93
*MIR.....	L 56
Mirroring of notes and text.....	S 49
mm>in.....	L 102
Model number, finding at boot-up.....	I 22
Modes.....	L 15
MODIFY.....	S 168-171
Modifying:	
Assembly.....	L 144
Drawing or instance.....	S 106-111
Macro file.....	U 62
Message file.....	U 93
Multiple components.....	L 54
Notes and text.....	L 196, L 251
Parts file.....	U 108
Parts screen menus.....	L 287
Process file.....	U 52
Tablet menu.....	L 287, U 85
Window.....	L 51

MOD_NOTE*	L 177, L 232
Monitor, setting default with codeword utility	I 46
Mouse	L 13
MOVE	S 172-173
*MOVE	L 56
MOVE*	L 70
Moving:	
Components	L 70
Connected part	L 193
Cursor	L 13
Cursor in text editor	M 32
Files	M 59, M 138
Parts list	L 164
Text in text editor	M 40, M 95

n

Naming conventions	U 36-37
Naming nets	L 195
Nesting level:	
Description	L 66, U 28-31, U 78
Setting with WINDOW	S 227
Net	U 126
Net element	U 126
Net name	U 126
Net-oriented connection list	U 148-152
Nets, naming	L 195
Netting level	U 128
NET_NAME	L 178, L 233
New	M 149
Newsysvol	M 78
NO FILL USE (Photoplot/Drill)	U 209
*NOLK	L 260
NOP	S 174
Note:	
ADD	S 48-52
Adding	L 94
Changing with MODIFY	S 169
Setting font size with FSIZE	S 122
Note string, modifying	L 196, L 251
Note stroke table, loading with LOAD NSTROKE	S 156

NOTE USE:

(Photoplot/Drill).....	U 209
Specifying aperture.....	U 198
NO_FILL identifier (Photoplot/Drill).....	U 198
*NUMB.....	L 148
Number of decimal places.....	S 78
Numeric operators, calculating with EVALUATE.....	S 116
Nut and washer items, adding.....	L 149

O

Operating system files.....	M 18
Operators, calculating with EVALUATE.....	S 116-117
Options, definition.....	U 2
Orientation of grid with GRID command.....	S 136
Origin placement.....	U 77
*ORTHO.....	L 82
Orthogonal axis.....	U 21-24
Outline.....	L 177
Outline, adding.....	L 195
Outlining a part.....	L 239
OVAL, ADD.....	S 40-42

p

\$P.....	L 17
PACK.....	L 80, S 175
Pad locations, adding.....	L 240
Pad set, adding.....	L 238
Pad, redefining.....	L 242
PAGE.....	S 176
Page.....	M 107
Panning with WINDOW.....	S 228
*PAR.....	L 28
Parameter:	
Modifying with EDIT.....	S 111
Values of macro instances.....	S 82

Part:

Aligning	L 185, L 242
Geometry, defining	L 183
Information, collecting	L 147
Modifying screen menus	L 289
Name	U 126
Outlining	L 239
Type, adding	L 186, L 243
Verification	L 189, L 245

Parts file:

Changing.	L 156
Editor	U 105
Listing.	U 109
Modifying.	U 108
Selecting.	U 105
Structure.	U 105

Parts list:

Creating.	L 159
Moving	L 164
Stepping border	L 165

PART_TYP	L 178, L 233
--------------------	--------------

Pascal 3.1:

Installing alongside HP EGS 2.1	I 70
Installing HP EGS on top of	I 67-69

Pascal command line	M 71
-------------------------------	------

Passive parts, connecting	L 182
-------------------------------------	-------

Passwords, defining	M 128
-------------------------------	-------

PAUSE.	S 177
----------------	-------

PAUSE ON ERROR (Photoplot/Drill)	U 207
--	-------

PC board	U 164, U 169
--------------------	--------------

Peripherals:

Bus addresses of	I 16-17
Select codes of.	I 16-17

Permanent	M 79
---------------------	------

*PERP	L 28
-----------------	------

Personality macros	S 6
------------------------------	-----

Personality, customized	U 100
-----------------------------------	-------

Photoplot and Drill post-processor files	M 25
--	------

PHOTOPLOT FORMAT (Photoplot/Drill).	U 208
---	-------

PHOTOPLOT ORIGIN (Photoplot/Drill)	U 210
--	-------

PHOTOPLOT ORIGIN, using.	U 187
----------------------------------	-------

PHOTOPLOT:

Description.....	U 171
Output.....	U 183
Photoplot post-processor:	
Codeword.....	U 174-178
Installing.....	U 174-178
Output.....	U 183-193
Overview.....	U 181-182
Using.....	U 179-180
PHOTOPLOT UNITS FACTOR (Photoplot/Drill).....	U 187, U 213
PHOTOPLOT/DRILL Commands.....	U 203-216
Physical board layers.....	L 227
Physical connection rules.....	U 127
Physical layer:	
Circuit side.....	L 269
Ground.....	L 270
Physical ports, adding.....	L 188, L 244
PHY_PORT.....	L 178, L 233
*PLMI.....	L 91
PLOT.....	S 178-186
PLOT*.....	L 38
Plotter:	
Pen number.....	U 52
Setting the bus address.....	I 17
Plotting:	
Drawing.....	L 38
Methods.....	L 38
Tablet menu.....	L 285, U 89-90
To a file.....	S 182
*PNTS (*3PNTS).....	L 28
POINT.....	S 187
*POL2.....	L 103
Polar copy, creating with STEP.....	S 203
POLYGON FILL USE:	
(Photoplot/Drill).....	U 210
Using tagged associated text.....	U 197
Polygon:	
ADD.....	S 44-45
Adding.....	L 32
Replacing portion with STRETCH.....	S 212

Port:	
Definition	U 126
Number, specifying with SHOW	S 202
On a part.	U 126, U 130
Position number of screen menus.	S 164
Post-processor preparation.	L 252
Post-processors	U 78
Power and ground layers	L 231
Power-up display	I 22
PREFIX	L 80, S 188
PREFIX (Photoplot/Drill)	U 210-211
Prefix	M 54, M 150
Prefix volume	L 7, M 6
Primary pads	L 280
Primitive:	
Adding.	L 23
Description	L 23, S 4
Primitive pads	L 279
:PRIMITIVE snapping mode	U 15
Printed circuit board connection lister	U 140, U 142
Printed Circuit Board Layout:	
Description	L 1
Library parts	L 278
Loading.	L 235
Personality	L 173, L 223
Tablet menu	L 284
Printing a drawing	L 38
Printing a text file	M 47
Process file:	
Archiving with a drawing	S 54-55
Building or modifying	U 52
Description	L 99, L 175, L 224
Layers	L 227
Listing contents with LIST	S 153
Loading with LOAD	S 156
Placing at the beginning of a GENERATE file.	S 131
Saving with SAVE	S 196-198
Setting resolution with RESOLUTION	S 192-193
Product number:	
Finding at boot-up.	I 22
Finding with the Codeword utility	I 45

Program:	
Loading with LOAD	S 156
User-written	U 46
Properties	U 130
Proportionally spaced characters.....	U 98

q

*QTY	L 148
Quit	M 108, M 152
QUIT	S 189
QUIT (Photoplot/Drill).....	U 215

r

\$R.....	L 17
*R&C	L 30, L 88
Radial dimensions, adding.....	L 123
Radial matrix, creating with STEP.....	S 204-207
Radius dimension	S 19
*RADs	L 102
RAM, seeing the amount of at boot-up.....	I 22
Rat's nest:	
Generating.....	U 164
Inputting traces.....	L 256
*RATO	L 102
:RAWPOINT snapping mode	S 4, U 15
Re-configuring HP EGS menus	M 17
Re-editing a drawing.....	L 248
Re-routing traces.....	L 258, L 260
RECALL.....	L 5
*RECT	L 103
Rectangle:	
ADD	S 46-47
Replacing portion with STRETCH.....	S 213
Recursive macros	U 67
Redefining a pad	L 242

Reference designator:

Adding	L 186, L 243
Description	U 126, U 129
Reference library parts	L 278
Refreshing text editor screen	M 31, M 120
REF_DESI	L 178, L 233
Relational operators, calculating with EVALUATE	S 117
REMOVE	L 62, L 80, S 190-191
Remove	M 153
Removing:	
Construction geometry	L 118
Files	L 62, M 61, M 153
Replace	M 37, M 111
Replacing strings in text editor	M 37, M 111
RESET (Photoplot/Drill)	U 211
RESOLUTION	S 192-193
RESOLUTION (archive)	S 288-289
RESOLUTION (Photoplot/Drill)	U 211
Resolution:	
Arc	S 11
Circle	S 15
Freehand line	S 34-37
Oval	S 41
Restarting last program or subsystem	M 84
Restricted components, copying	L 45
RETRIEVE	L 77, S 192
REVISION	S 193
*Right30	L 82
Root directory	M 8, M 9
*ROT	L 56, L 74
Rotating components with MODIFY	S 169
Rotation of notes and text	S 49
ROUTE*	L 177, L 232, L 260
Rplace	M 37, M 111
Rule Check post-processor files	M 24
Run	M 80
Run time:	
Estimation for IGES Translator	U 249, U 257
Parameters	U 93-94

S

Save	M 155
SAVE	S 196-198, U 52, U 62
SAVE (archive)	S 290
Saving a drawing	L 36, L 45, U 52, U 62
*SCAL	L 56
SCALE	L 38
Scale factor, changing with MODIFY	S 169
Schematic:	
Library parts	L 218
Tablet menu	L 222
Schematics	U 161
Scientific functions, calculating with EVALUATE	S 117
Screen dump:	
Initiating with PAGE	S 176
With PLOT	S 181
Screen menu:	
Building	U 57
Description	L 15, U 7
File structure	U 58
Modifying with MENU	S 164-166
Saving with SAVE	S 196-198
SEARCH	S 200-201, U 50
Search file:	
Building	U 50-51
Listing contents with LIST	S 153
Loading with LOAD	S 156
Saving with SAVE	S 196-198
Searching for patterns in text editor	M 42, M 99
Secondary commands	U 10-12
Select	L 23
Select codes, setting on a peripheral	I 16-17
Selecting apertures with PHOTOPLOT	U 188-193
Selecting menu items	L 16
*SEMI	L 40
Semicircle and tangent lines, adding	L 106
Serial number, finding with the Codeword utility	I 45
Set	M 43, M 115
Shape of notes and text	S 51

Shape, stretching	L 32
Shared Resource Manager, see SRM	I 61-63
shON/shOFF	L 44
Short macros vs. long macros	U 63
SHOW	S 202, U 5
SHOW (archive)	S 291
SHOW*	L 44, L 53
Slant of notes and text	S 49
*SLNT	L 74, L 148
SMASH	S 203
SMASH identifier	U 129, U 160
Smashing an instance	L 69
Smashing level, connection listing	U 128
Snapping mode:	
Changing during command	S 4
Definition	L 17
Example of changing during command	S 26
:GRID	S 4, U 15
:INTERSECT	S 4, U 15
:PRIMITIVE	S 4, U 15
:RAWPOINT	S 4, U 15
Selecting	S 4
Selecting points while changing	S 4
:VERTEX	S 4, U 15
*SNGL	L 91
Solder resist, adding	L 227
SORT APERTURE (Photoplot/Drill)	U 212
SORT DRILL (Photoplot/Drill)	U 212
SORT PROXIMITY (Photoplot/Drill)	U 212
SOURCE (Photoplot/Drill)	U 213
Source voltage input, adding	L 194
*SPAC	L 41
Space planning personality	U 39-100
Special reference designators	U 129
SPOOL	L 38
sPT = 1/sPT = 2	L 44
SRM:	
Compared to local disc notation	U 37
Changing the working directory	M 69, M 150
Creating directories	I 63

Directory structure	I 61
Explanation	M 8
File organization	M 8
Listing directories.	I 62
Making files and directories.	M 69, M 146
Making new pointers to a file	M 69, M 133
Pathnames	M 9
Units and volumes	M 10
Start file:	
Building	U 48
With HP EGS Graphics Editor.	U 49
STEP	S 204-207
Step specification:	
Adding with ADD	S 7-8
Example of	S 47
Stepping parts list border	L 165
Stick characters (note)	S 48-52
STOP key.	L 6
Stopping an operation	L 79
Stream.	M 81
Stream files	L 78
Stream file, executing	M 81
Streaming files with INPUT	S 148
STRETCH.	S 208-216
STRETCH*	L 258
Stretching:	
Center line	L 142
Re-routed trace.	L 261
Shape	L 32
String functions, calculating with EVALUATE	S 117
Stroke tables:	
Creating	U 97
Format	U 97
For notes and text	S 51
Loading with LOAD	S 156
Troubleshooting.	U 99
Used in connection listing.	U 147
*STYL	L 41
Style, hatching.	S 25
Suffixes of files.	S 65

Suffixes of HP EGS files	U 36-37
Summagraphics digitizer	I 20-21
Surface finish symbol, adding	L 128
Symbolic layer:	
Defining with EQUATE	S 113
Type	U 19
Symbolic mode, setting display with WINDOW	S 229
Syntax diagrams, interpreting	S 1
SYSTEM directory, contents of	I 61
System volume:	
Definition	M 6
Designating new system volume	M 78
System-defined macro constants	U 68-73
System:	
Constants	U 68-70
Cursor	S 72-75
Grid	L 17, U 12-13
Grid points	U 13-14
Grid points, assigning distance with UNITS	S 222-223
SYSTEMS directory, content of	I 61

t

Tablet menu:	
Activating	L 285, U 90
Building	U 85-91
Description	U 7
Digitizing with	S 60-61
File	U 90
How the system interprets menu items	U 91
Loading with LOAD TMENU	S 156
Mechanical Drafting	L 172
Modifying	L 287
PC Board Layout	L 284
Plotting	L 285, U 89
Saving with SAVE	S 196-198
Schematic	L 222
Setting the bus address	I 17
Toggling with TMENU	S 217

Tag:	
Field	U 93
Value	U 21, U 136-137, U 149
*TAN1	L 28, L 30
*TAN2	L 28, L 30, L 40, L 89
Temporary macros	U 66
Terminating a primary command	S 5
Terminating commands	L 25
Text editor:	
Aborting commands	M 31
Adding text	M 29, M 102
Adjusting margins	M 45, M 93
Confirming commands	M 31
Copying text	M 40, M 95
Deleting text	M 34
Filling paragraphs	M 46, M 106
Formatting text	M 43, M 115
Moving text	M 40, M 95
Moving the cursor	M 32
Printing a file	M 47
Refreshing screen	M 31, M 120
Replacing character strings	M 37, M 111
Searching for patterns	M 42, M 99
Typing over characters	M 36, M 121
Using	M 27
Using to define macros	S 82
Text:	
ADD	S 48-52
Changing slant with MODIFY	S 169
Changing with MODIFY	S 169
Loading stroke table with LOAD TSTROKE	S 156
Setting font size with FSIZE	S 122
Setting slant with TSLANT	S 220
Time:	
Entering	L 8
Setting system time	M 85
Title block, updating	L 129, L 153
TMENU	S 217, U 86, U 91
TNAME	S 218
TNAME (archive)	S 292

Tolerance values	S 18-22
Tooling marks, adding	L 262
*Top30.....	L 82
Traces:	
Adding.....	L 261
Connecting	L 257, L 271
Tracing or flashing	U 181
Transferring files:	
From magnetic/paper tape	M 16
To magnetic/paper tape	M 16
Transformations with PHOTOPLOT	U 186
Translate	M 61, M 157
TSLANT	S 220
TSLANT (archive)	S 293
Type-ahead buffer	U 46
Typing over characters in text editor.....	M 36, M 121

u

UNDO*	L 23
Unfilling an item.....	L 50
Ungathering a collection.....	S 123-126
Unit directory	M 159
Unit number (disc drive number)	I 5
Units	M 5
UNITS	S 222, U 12-13, U 53
UNITS (archive)	S 294-295
Unused space on a disc	S 63
Unwrapping instances with SMASH.....	S 203
Updating a title block	L 129, L 153
User restart	M 84
User:	
Grid.....	L 18, U 14-15
Units.....	U 12-15, U 53
Written applications.....	U 46

V

Variables, calculating with EVALUATE	S 116
Verify	M 120
Verifying the part	L 189, L 245
Version	M 85
VERSION (Photoplot/Drill)	U 214
:VERTEX snapping mode	U 15
Vertex:	
Adding	L 60
Adding with STRETCH	S 208-216
Snapping mode	S 4
Vertical dimension:	
Adding	L 126
Line	S 19
Vertices, adjusting with a GATHER collection	S 123-128
VIA identifier	U 129, U 160
VIAs	L 276
Video interface card	I 18-19
Video output with DISPLAY	S 100-101
VIEW (Photoplot/Drill)	U 214
View, copying	L 133
Viewing instances	L 182
Viewport:	
Description	U 4-6, U 59
Explanation	L 12, L 71
Specifying number with WINDOW	S 228
Specifying with SHOW	S 202
VIRTUAL DRILL (Photoplot/Drill)	U 215
VIRTUAL PHOTOPLOT (Photoplot/Drill)	U 215
VLCTY	L 38
Volume search table	M 9
Volume:	
In search table	S 200-201, U 50-51
Name	L 37
Packing with PACK	S 175
Prefix	L 9
Specifying default volume with PREFIX	S 188
VOLUMES	S 224

Volumes:

Changing the default volume	M 54, M 150
Concepts	M 3
Listing	M 53, M 161
Names	M 4
Removing files	M 61, M 153
Storage capacity	M 7
\$VR.	L 17
*VRTX	L 259

W

Washer and nut, adding	L 136
*WIDTH	L 26, L 260
What	M 87, M 162
WHEEL (Photoplot/Drill).	U 216
Wheel File:	
Aperture entries	U 199
Aperture height	U 200
Aperture name	U 199
Aperture number	U 199
Aperture shape	U 199
Aperture type	U 200
Aperture width	U 200
Width:	
Changing	L 51
Changing with MODIFY	S 169
WIND* PAN.	L 54
WIND* REDRW	L 31
WINDOW	S 226-229, U 5, U 19, U 78, U 92
Workfiles:	
Clearing	M 66, M 149
Copying a file into	M 63, M 141
Copying to a file from	M 63, M 141
Displaying name	M 66, M 162
Displaying state	M 66, M 162
Executing	M 80
WRAP	S 230-233
WRAP*	L 64

X

X,Y location errors, correcting	L 26
Xchg.	M 36, M 121

Z

Zap.	M 35, M 123
Zero.	M 163
Zero-base datum dimension	S 19
ZONE_ON	L 103
Zoom factor, setting with WINDOW	S 227
Zooming in on an item	L 35



Reorder Number
98305-90008

Printed in U.S.A. 7/85



98305-90600
Mfg. No. Only