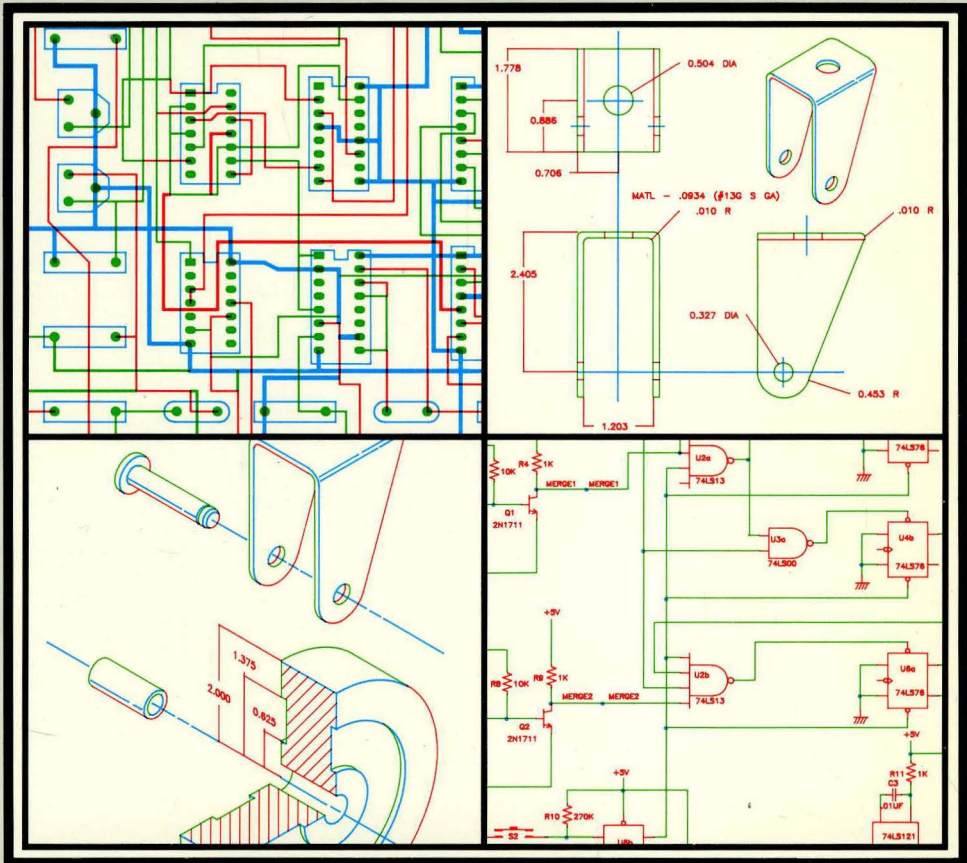


# Understanding HP EGS 2.1



# **Understanding 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

Using This Manual . . . . .	ix
Overview of the HP EGS System . . . . .	ix
Where To Go From Here . . . . .	x
<b>Chapter 1: The Graphics Editor</b>	
Introduction . . . . .	1
Overview of the Graphics Editor . . . . .	2
Interacting with the HP EGS Graphics Editor . . . . .	4
The Viewports . . . . .	4
Screen and Tablet Menus . . . . .	6
Alpha and Graphics Screens . . . . .	7
Command Entry . . . . .	8
Types of Commands . . . . .	10
The HP EGS Grid System . . . . .	12
Selecting Drawing Points with Snapping Modes . . . . .	15
Dynamic Component Tracking . . . . .	16
Evaluation of Mathematical Expressions . . . . .	16
HP EGS Drawings . . . . .	17
Drawings and Drawing Files . . . . .	17
Drawing Components . . . . .	18
Layers . . . . .	18
Associated Text . . . . .	20
Isometric Drawing . . . . .	21
Instances and Nesting Levels . . . . .	28
Logical Levels . . . . .	31
HP EGS Commands . . . . .	32
<b>Chapter 2: Customizing Your System</b>	
Introduction . . . . .	35
Naming Conventions . . . . .	36
Defining the Personality . . . . .	38
Space Planning Example . . . . .	39
Making a Volume or Directory . . . . .	41
Space Planning Example . . . . .	41



Building a Control File . . . . .	42
Control File Structure . . . . .	42
Using the INPUT Command in Control Files . . . . .	46
User-written Applications. . . . .	46
Space Planning Example. . . . .	47
Building a Start File . . . . .	48
Space Planning Example. . . . .	48
Building Volume Search Tables . . . . .	50
Space Planning Example. . . . .	51
Building or Modifying a Process File. . . . .	52
Defining Layers . . . . .	52
Specifying Default Values . . . . .	53
Space Planning Example. . . . .	54
Building or Modifying a Screen Menu . . . . .	57
Screen Menu File Structure. . . . .	58
How The System Interprets Screen Menu Items. . . . .	59
Space Planning Example. . . . .	60
Building or Modifying a Macro File. . . . .	62
Defining a Macro . . . . .	62
Short Macros vs. Long Macros . . . . .	63
How the System Interprets Information. . . . .	64
Macro Substitutes for Commands . . . . .	65
Temporary Macros. . . . .	66
Macros to Save Values and Set Modes . . . . .	67
Macros to Modify Existing Information. . . . .	68
System-defined Macro Constants. . . . .	68
Conditional Macros . . . . .	75
Space Planning Example. . . . .	76
Creating Library Parts . . . . .	77
Special Drawing Considerations. . . . .	77
Macro Instance Library Parts . . . . .	78
Space Planning Example. . . . .	83
Building or Modifying a Tablet Menu. . . . .	85
Drawing a Tablet Menu. . . . .	87
Plotting a Tablet Menu . . . . .	89
Activating a Tablet Menu. . . . .	90
How the System Interprets Menu Items. . . . .	91
Space Planning Example. . . . .	91

Modifying the Message Files .....	93
Structure of a Message File .....	93
Runtime Parameters .....	93
Editing and Indexing a Message File .....	94
Examples of Message Files .....	95
Help Messages for Personalities .....	96
Creating New Stroke Tables .....	97
The Stroke Table Format .....	97
Example .....	98
Troubleshooting .....	99
Putting the Personality Together .....	100
Space Planning Example .....	100

### **Chapter 3: Material Listing**

Introduction .....	103
The LIST MATERIAL Command .....	104
Drawing Conventions .....	104
LIST MATERIAL Output File .....	104
The Parts File Editor .....	105
Selecting a Parts File .....	105
Parts File Structure .....	105
Modifying a Parts File .....	108
Listing a Parts File .....	109
The Electrical Engineering Material List Post-Processor .....	110
Generating a Post-Processed Material List .....	112
Material List Error Checking .....	114
The Mechanical Engineering Material List Post-Processor .....	115
Material List Output Format .....	117
Generating a Post-Processed Material List .....	120
Listing a Post-Processed Material List .....	122
Material List Error Checking .....	123

## Chapter 4: Connection Listing

Introduction .....	125
Connection Listing Concepts .....	126
Definition of Terms .....	126
Logical vs. Physical Connection Listing .....	127
Connection Hierarchy .....	127
Special Reference Designators .....	129
Bus Notation .....	129
Properties .....	130
Implicitly Stepped Components .....	130
Installing the Connection Lister .....	131
Local Installation .....	131
SRM Installation .....	132
Installing Your Codeword .....	132
Using the Connection Lister .....	136
Using Associated Text .....	136
Archiving Drawings .....	137
Entering the Connection Lister Program .....	137
Running the Connection Lister .....	139
Connection Lister Default Files .....	142
Connection Lister Message File .....	146
Efficient Processing of Connection Lists .....	147
Connection Lister Output File .....	148
Connection Lister Operational Errors .....	156
Connection Lister Error File .....	157
Customizing the Connection Lister .....	159
Connection List Utilities .....	161
Merging Connection Lists .....	161
Generating a Rat's Nest .....	164
Comparing Connection Lists .....	169

**Chapter 5: Photoplot and Drill Post-Processors**

Introduction .....	171
Installing the Photoplot and Drill Post-Processors .....	172
Local Installation .....	172
SRM Installation .....	173
Installing Your Codeword .....	174
Obtaining Your Codeword .....	176
The Codeword Installation Program .....	178
Using the Photoplot and Drill Post-Processors .....	179
Operating from the HP EGS Manager Menu .....	179
Operating from the Pascal Command Line .....	180
Overview of PHOTOPLOT .....	181
PHOTOPLOT Output .....	183
Overview of DRILL .....	194
DRILL Output .....	194
Using Tagged Associated Text .....	197
The Wheel File .....	199
The Default File .....	201
Example Default File .....	202
PHOTOPLOT/DRILL Commands .....	203
Customizing the Message File .....	216
Creating Library Parts .....	219
Software Problems .....	225

**Chapter 6: IGES Translator**

Introduction . . . . .	229
Installing the IGES Translator . . . . .	231
Local Installation . . . . .	231
SRM Installation. . . . .	231
Installing Your Codeword . . . . .	232
Obtaining Your Codeword . . . . .	234
The Codeword Installation Program . . . . .	236
IGES File Format . . . . .	237
HP EGS - IGES Relationships . . . . .	238
The EGS-to-IGES Translator . . . . .	239
Archiving Drawings . . . . .	239
EGS-to-IGES Translator Default File . . . . .	240
Using the EGS-to-IGES Translator . . . . .	244
EGS-to-IGES Translator Messages. . . . .	247
Limitations of the EGS-to-IGES Translator. . . . .	248
Estimating Mass Storage Requirements and Run Time . . . . .	249
The IGES-to-EGS Translator . . . . .	250
Retrieving Drawings. . . . .	250
IGES-to-EGS Translator Default File . . . . .	251
Using the IGES-to-EGS Translator . . . . .	253
IGES-to-EGS Translator Messages. . . . .	254
Limitations of the IGES-to-EGS Translator. . . . .	256
IGES-to-EGS Translator Process File. . . . .	257
Estimating Mass Storage Requirements and Run Time . . . . .	257

**Application Notes**

## Using This Manual

Welcome to *Understanding HP EGS*! This manual is designed to give you an in-depth look at how your HP EGS system works. Throughout its chapters you will learn about:

- Using the Graphics Editor
- Customizing the system to meet your needs
- Creating material and connection lists
- Creating output to run photoplotters and N/C drill machines
- Translating drawings to and from other systems using the IGES Translator

The discussions in this manual assume you are familiar with the lessons presented in *Learning HP EGS*; if not, you probably will want to read them first and get comfortable with the “basics.”

To help you understand the system, let’s look at the individual parts of the HP EGS system, and at how the menu selections available to you relate to the HP EGS manual set.

### Overview of the HP EGS System

The HP EGS system contains several programs that allow you to produce drawings and manage files. These programs could be classified as follows:

Manager	The Manager program is the one you enter every time you use HP EGS. It manages and controls all the programs you might run. The Manager displays menus from which you select menu items. When you request a program from a menu, the system automatically loads all of the necessary files and commands for that program. The Manager’s menus are not fixed – you may change them to run user-defined programs directly from the Manager, if desired.
Graphics Editor	The Graphics Editor program is the heart of the HP EGS system. It allows you to create and modify drawings, just as a text processor allows you to create and modify textual material. You can create new drawings, modify them, save them for later use, plot them, and so on. The Graphics Editor responds to a special set of commands that is fully defined in the <i>HP EGS Syntax Reference</i> .

- Personalities** You may alter the appearance of the Graphics Editor by creating screen and tablet menus, command macros, and parts libraries, and by “customizing” the drawing environment. Personalities are collections of such customized files. HP EGS provides four personalities for your use – you may modify these or create others to tailor HP EGS for your specific application.
- Pre- and Post-Processors** These programs provide additional functionality to the system. They produce information that can be used to create a drawing or derive additional data from a drawing. For example, the Rat’s Nest program allows the designer of a printed circuit board to automatically produce an “airline routing” of the various connections on a drawing. This drawing is produced from a connection listing and is an example of pre-processing since the data is used to create a new drawing. The Photoplot program, on the other hand, is a post-processor which takes as input a completed drawing and produces output that drives a photoplotter. You may write your own pre- and post-processors to extend the HP EGS system to handle your particular applications.
- Utilities and Pascal Programs** These programs are not involved in the creation of drawings, but they let you create and manipulate files, and initialize and backup discs. Specifically they help you move data among various media, convert characters to other codes, and break-up and merge large files.

## **Where To Go From Here**

The following diagram shows all the menu selections you see when running HP EGS and where, in the HP EGS manual set, you can find information on each menu item. For most subjects, you may jump right to the chapter(s) shown here for detailed information – it is not necessary to read the chapters consecutively.

# THE HP EGS SYSTEM

## MAIN MANAGER MENU

## SECONDARY MANAGER MENU

## PROGRAM DESCRIPTION

## MANUAL REFERENCE

## OTHER TOPICS OF INTEREST

System Utilities
General Drawing Editor
Electrical Engineering
Mechanical Engineering
IGES Translator (Optional)

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

Initializes mass storage media  
 Copies an entire CS80 disc to internal tape  
 Creates and modifies text files  
 Manipulates files and volumes  
 Provides menu of PASCAL commands  
 Segments large files into multiple smaller files  
 Reconstructs segmented files into one  
 Transfers files to magnetic and paper tape  
 Converts characters in a file (e.g., ASCII to EBCDIC)  
 Checks and indexes message files  
 Installs all HP EGS codewords  
 Powers-up the system  
 Recognizes flexible discs placed in drives

Managing and Editing Files with HP EGS  
  
 Understanding HP EGS—Ch 2  
 Installing HP EGS  
 Managing and Editing Files with HP EGS

RETURN to Main Menu
Printed Circuit Board Layout Editor
Electrical Schematic Drawing Editor
EE Parts File Editor
EE Material Lister
Convert Old EE Parts Files
Printed Circuit Board Connection Lister
Electrical Schematic Connection Lister
Compare Connection Lists
Merge Connection Lists
Generate Rat's Nest
PHOTOPLOT/DRILL Post-Processor (Optional)

Loads Graphics Editor and General Drawing personality  
  
 Loads Graphics Editor and PC Board Layout personality  
 Loads Graphics Editor and Schematic Drawing personality  
 Edits parts files used by the EE Material Lister  
 Outputs a parts list from PC board and schematic drawings  
 Converts EGS/200 parts files to HP EGS 2.1 format  
 Creates connection lists  
 Compares two connection lists and reports differences  
 Merges multiple connection list output files by net name  
 Draws airlines on a placed PC board  
 Outputs PC board data to drive a photoplotter and N/C drill

Learning HP EGS—Ch 3-5  
  
 Learning HP EGS—Ch 9  
 Learning HP EGS—Ch 8  
 Understanding HP EGS—Ch 3  
 Understanding HP EGS—Ch 3  
 Understanding HP EGS—Ch 3  
 Understanding HP EGS—Ch 4  
 Understanding HP EGS—Ch 4  
 Understanding HP EGS—Ch 4  
 Understanding HP EGS—Ch 4  
 Understanding HP EGS—Ch 5

RETURN to Main Menu
Mechanical Drafting Editor
ME Parts File Editor
ME Material Lister

Loads Graphics Editor and Mechanical Drafting personality  
 Edits parts files used by the ME Material Lister  
 Outputs a parts list from mechanical drawings

Learning HP EGS—Ch 6 & 7  
 Understanding HP EGS—Ch 3  
 Understanding HP EGS—Ch 3

RETURN to Main Menu
EGS to IGES
IGES to EGS

Translates an HP EGS Archive file into an IGES file  
 Translates an IGES file to an HP EGS Archive file

Understanding HP EGS—Ch 6  
 Understanding HP EGS—Ch 6

Installing Your System      Installing HP EGS  
 The Graphics Editor      Understanding HP EGS—Ch 1  
 Customizing Your System    Understanding HP EGS—Ch 2



# The Graphics Editor

## Chapter 1: The Graphics Editor

Introduction.....	1
Overview of the Graphics Editor.....	2
Interacting with the HP EGS Graphics Editor.....	4
The Viewports.....	4
Screen and Tablet Menus.....	6
Alpha and Graphics Screens.....	7
Command Entry.....	8
Types of Commands.....	10
The HP EGS Grid System.....	12
Selecting Drawing Points with Snapping Modes.....	15
Dynamic Component Tracking.....	16
Evaluation of Mathematical Expressions.....	16
HP EGS Drawings.....	17
Drawings and Drawing Files.....	17
Drawing Components.....	18
Layers.....	18
Associated Text.....	20
Isometric Drawing.....	21
Instances and Nesting Levels.....	28
Logical Levels.....	31
HP EGS Commands.....	32



# The Graphics Editor

Chapter

**1**

## Introduction

This chapter provides you with an understanding of the HP EGS Graphics Editor. The Graphics Editor is the program that allows you to create and modify drawings. In essence, it is the heart of HP EGS.

In this chapter you will recognize many terms and commands discussed in *Learning HP EGS*. The personalities described there are all built upon this underlying Graphics Editor. This chapter attempts to give you a better understanding of how the Graphics Editor works. The chapter discusses in detail:

- The viewports, menus, alpha and graphics screens, edit key functions, grid settings, and command entry.
- Drawings and drawing files.
- Components and how they can be used to make drawings.
- Using layers, nesting levels, logical levels, and associated text in your drawings.
- Isometric drawing.
- Graphics Editor commands, which you can use to create and edit drawings, gain information about drawings, change the drawing environment, name and manage files, input and output drawing information, create new commands, and customize the menus.

## Overview of the Graphics Editor

This chapter focuses on the Graphics Editor part of the HP EGS system. It is the program you will interact with most often. Because the Graphics Editor is used to produce drawings, most of the interaction with the program is by graphical images on the computer screen. Thus, when you perform an operation, the result of that operation is reflected in the image you see.

The Graphics Editor is command-driven. This means that you issue commands that tell it to perform actions. You can think of these commands as sentences in the language that HP EGS understands. This language is fully defined in the *HP EGS Syntax Reference*. In many ways we can draw parallels between English sentences and the commands that you give the system. For example, in the English language a command might look like this:

**verb – object – prepositional phrase – period** , where

- verb** is the command you want obeyed.
- object** is the recipient of the action.
- prepositional phrase** contains words used to more completely describe the action.
- period** is the English way of terminating the sentence.

The HP EGS commands are quite similar in that they look like this:

**command – component – options – semicolon** , where

- command** is the action that you want to perform. Commands are simple English verbs like ADD and MOVE that denote the desired operation.
- component** is the object you wish the command to operate on. For example, when you want to add something to a drawing, the component is the type of object you wish to add (circle, rectangle, etc). In some cases (like MOVE), the object of the action is specified by simply pointing at it on the screen.
- options** are additional information about the operation to be performed. For example, there are several ways to add circles to a drawing. The option describes which method to use. These options appear as one or more characters preceded by a colon.
- semicolon** is the equivalent of the English period. It is used to finish the command.

If you think of each HP EGS command as a sentence and try to understand the logic of the command, you can quickly learn the commands and how they are used.

There are many ways to give commands to the system. Each produces identical results. The various methods can be intermixed as needed although you will find that different methods are best suited to different situations. The methods of giving commands are:

- |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Keyboard              | You may type all the information that is needed using the keyboard. While it is often convenient to type in all of the information, you may find it easier to select points with a stylus and a graphics input device.                                                                                                                                                                                                                          |
| Graphics Input Device | You may select screen and tablet menu items and the position of objects in a drawing with a tablet, mouse, arrow keys, or digitizer.                                                                                                                                                                                                                                                                                                            |
| Screen Menu           | You may enter information into the system by selecting items from a screen menu. This is done using the graphics input device to position the cursor over the desired menu box and select the item. The information associated with the selected item is entered into the system just as if you had typed it in. This makes it convenient to issue commands by selecting a sequence of menu items that describe the action you wish to perform. |
| Tablet Menu           | You may enter information into the system by selecting items from a tablet menu. This is done by positioning the graphics tablet stylus over the desired menu item and selecting the item. The information associated with the selected item is entered into the system just as if you had typed it in. The tablet menu also has an area designated as the screen, from which you may select screen menu items and drawing point locations.     |
| Files                 | You can anticipate many sequences of commands that you will use on a daily basis. For example, you will likely define a standard drawing environment and use this environment (defined by a series of commands) over and over again. By placing these commands in a file, you can use them repeatedly without having to retype them. Issuing a single command to read an entire file of commands can save you considerable work.                |

## Interacting with the HP EGS Graphics Editor

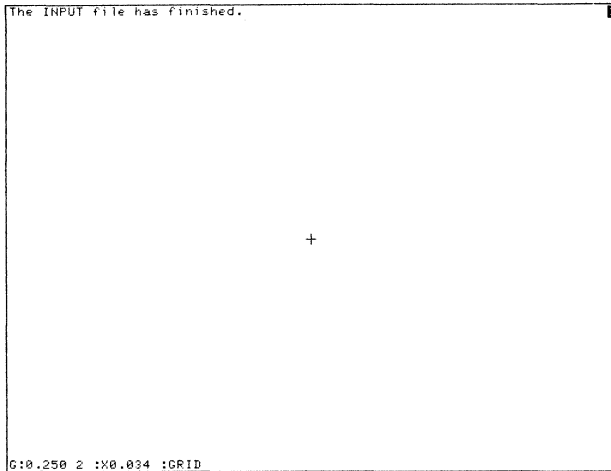
This section describes the Graphics Editor features that help you create drawings. It discusses in detail how to use the system's viewports, screen menus, and alpha and graphics screens; and how to enter commands, set grids, and select drawing points.

### The Viewports

A viewport provides a view into the drawing you are currently editing. The Graphics Editor can have one or two viewports.

When you first enter the Graphics Editor, there is no screen menu. As a result, there is only one viewport. If a screen menu is loaded into the system (as is the case with any of the HP EGS personalities), the system attempts to create the second viewport in the space left over by the menu.

When there are two viewports, they are numbered 1 and 2. The viewport occupying the major portion of the screen has the number 1 in its upper righthand corner. The smaller viewport has the number 2 in its upper righthand corner. Below are three examples of screens: one without a screen menu, one with a vertical menu and two viewports, and one with a horizontal menu and two viewports.









Screen menus can be either vertical or horizontal. In addition, they may be placed in any corner of the screen.

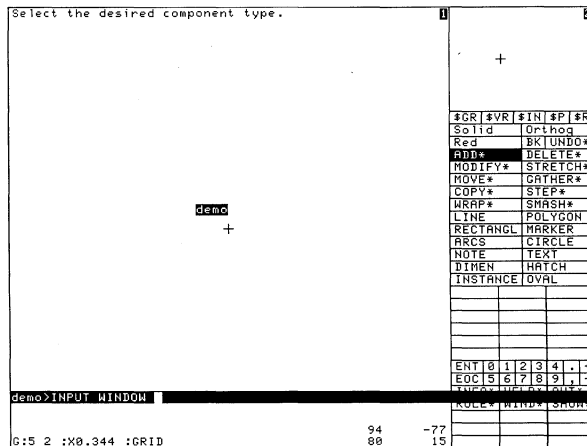
HP EGS can also recognize a tablet menu. This menu may be used in conjunction with a screen menu, if desired. It may contain any information that you would like passed to the Graphics Editor for action. Tablet menus are often desirable if you want a larger drawing area on your screen or of you want to be able to select from both textual and graphical representations of items.

For more information on how menus may be changed, please refer to Chapter 2: *Customizing Your System* and the MENU and TMENU commands in the *HP EGS Syntax Reference*.

### Alpha and Graphics Screens

The internal CRT on HP Series 200 Model 216, 217, 226, and 236 computers and Series 300 Model 310 and 320 computers can display two sets of images: alpha and graphic. If you have only an internal display, the system will switch from graphics to alpha whenever **f6** (the **ALPHA** softkey) is pressed. For the HP Model 220 computer, a separate graphics and alpha display is required to use HP EGS.

When you type from the keyboard, your input is displayed on an inverse video line near the bottom of the graphics display. You may type up to 255 characters on this line. As soon as you press **Return** to accept the input, the inverse video line disappears and the entire graphics display is visible.



## 8 The Graphics Editor

If the results of the typed-in command affect the contents of the viewport, the results will be seen graphically. For example, if you add an arc by typing in

```
ADD A4 :R20 :A 50 120 240 100,200 ;
```

from the keyboard, the input is printed on the inverse video line, but the arc appears on the graphics screen. (The complete syntax for the ADD ARC command is explained in the *HP EGS Syntax Reference*.)

### Command Entry

You may enter Graphics Editor commands in four ways: by typing them in on the keyboard, by selecting them with the graphics input device from the screen menu, by selecting from a tablet menu, or by inputting a stream of commands from a file.

A graphics input device allows you to see the effect of any command as it is being entered. To use the graphics input device, position the cursor over the menu command that you want to enter, and press.

A single command may be entered using both the graphics input device and the computer keyboard. When entering information from the computer keyboard, you may wish to switch at some point to the graphics input device. For example, you might type:

```
ADD PART :R90 Return
```

and then use the graphics input device to select a location in the drawing area. To do this, you would press **Return** when you finished typing and pick a location with the stylus. To switch from the graphics tablet to the computer keyboard (i.e., to type a drawing name or the characters in a note), you would simply start typing when the system asks for the character information.

You may enter parameters associated with a command by selecting the appropriate item from a menu slot (i.e., entering the layer number or the angle of rotation from the numeric column) or by typing in information from the keyboard.

You must enter a minimum of three characters of a command name or as many as are needed to make the command unique. For example, entering WIN is equivalent to entering the entire command name WINDOW. Additional characters of the command name may be entered, but they must be the correct characters in the correct order. For example, WIN, WIND, and WINDO are all correct entries for the WINDOW command. However, WINDW is not a correct entry for the WINDOW command because the last character W must have the character O before it.

The system recognizes spaces, commas, and **Return** as delimiters. They should be used to separate adjacent parameters (e.g., :L :F or :L :F , 100,100 or 100 100). If these delimiters are omitted when required, an error will occur.

A command must be completed before the system executes it. Graphics Editor commands usually end with either a semicolon or a primary command (discussed in more detail later in this chapter). In some cases double picking the same point completes the command. When using the keyboard, always press **Return** after each command. This tells the system that you are finished typing the command, and allows input to continue from the graphics tablet. Do not use **Select** or **f4** (the **CONT** softkey) to end a command.

### Special Editing Keys

Certain functions can be performed using dedicated keys on your keyboard. Note that not all of these keys are available on every keyboard. The features implemented with dedicated keys are:



moves the cursor to the left one character at a time while not erasing the characters. **Shift** jumps the cursor to the first character of the line.



moves the cursor one character at a time to the right. **Shift** jumps the cursor to the end of the line.

**Backspace**

is the equivalent of the when the cursor is anywhere but at the end of the line. This key simply moves the cursor to the left. At the end of the line, it erases each character as it moves the cursor to the left.

**Insert char**

changes the cursor to a block character (inverse video) and allows the insertion of characters. If pressed a second time, it returns to the normal mode. The effects of this key are cancelled if the cursor is moved to the end of the line. The key has no effect if the cursor is already at the end of a line.

**Delete char**

deletes the character underlined by the cursor.

### CLR-END

clears the line from the cursor to the end of the line, while leaving the cursor in its current position. This key also cancels the insert mode if it is set. On the HP 46020A keyboard, the **f3** key performs this function.

**Clear line**

clears the entire line leaving the cursor at the beginning of the line. It cancels the insert mode if set.

### RECALL

clears the line and recalls the previous entry. On the HP 46020A keyboard, the **f2** key performs the recall function. Additional presses recall the next most recent entry until the end of the buffer is reached. To move through the recall buffer forward rather than backwards, use **Shift** **f2** (unless you have an HP 9920 with a small keyboard, in which case use **CTRL** **f2**). A beep sounds whenever you reach either end of the buffer during the recall process.

- Stop** stops the output of the PLOT, ARCHIVE, LIST, CATALOG, and GENERATE commands. **Stop** used during the WINDOW and SHOW commands will terminate the drawing of the entire display.
- ALPHA** turns on the alpha display. On the HP 46020A keyboard, the **f6** key performs this function.

If you are using the type-ahead buffer, which allows you to enter commands while the system is processing the current command, several other keys are of interest:

- Backspace** prints a <sup>b</sup>s on the screen and negates the character to its left in the buffer.
- CTRL** **Backspace** removes the previous character from the buffer and the screen display.
- Clear line** inserts a block character that clears the type-ahead buffer line.
- CTRL** **Clear line** removes the line of commands from the type-ahead buffer and the screen.

The KNOB (on HP 98203A keyboards) and arrow keys can also be used to move the alpha and graphics cursors. After moving the graphics cursor with the knob, you must press **Return** to return cursor control to the graphics tablet.

### Types of Commands

The Graphics Editor recognizes two types of commands: primary and secondary. In general, primary commands alter the drawing data itself. These commands are used to create and modify drawings. A primary command may be terminated only by another primary command. The following commands are primary commands:

ADD	GENERATE	REMOVE
ARCHIVE	GROUP	RETRIEVE
COPY	LIST	SAVE
DELETE	LOAD	SMASH
EDIT	MODIFY	STEP
EXIT	MOVE	STRETCH
GATHER	NOP	UNDO
	QUIT	WRAP

Secondary commands do not alter the drawing data. These commands change the drawing environment and obtain information about the system. Entering a secondary command does not affect the primary command in use. A secondary command is terminated either with a semicolon or if interrupted by any other command. The following commands are secondary commands:

AREA	DISPLAY	MENU
BACKUP	DISTANCE	PACK
CALIBRATE	DUMP	PAGE
CATALOG	ECHO	PAUSE
CHANGE	EQUATE	PLOT
CURSOR	EVALUATE	POINT
DATUM	FILECOPY	PREFIX
DECIMALS	FSIZE	RESOLUTION
DEFINE	GRID	REVISION
DIM_DEC	HELP	SEARCH
DIM_DIR	IDENTIFY	SHOW
DIM_EXTEND	INAME	TMENU
DIM_FSIZE	INPUT	TNAME
DIM_RADIX	LEVEL	TSLANT
DIM_SCALE	LOCATOR	UNITS
DIM_TFSIZE	LOCK	VOLUMES
DIM_TSLANT	MEMORY	WINDOW
DIM_UNITS		

Whenever a primary command is entered, it becomes the current command. The system remembers the command and any parameters and data entered with it. The exception to this rule is UNDO, which removes the current command from memory. To enter a primary or secondary command again using slightly different parameters, you need only enter those parameters and data that change. For example, suppose that you wish to add two text components to your drawing. To add the first component you would enter:

```
ADD T1 :F 100 :R 90 "This is Text" 50,50 ;
```

This adds the text to the drawing with a font size of 100 user units and a rotation of 90 degrees at the location 50,50. To add the second text component with the same angle of rotation but a font size of 200 and a different location, it is only necessary to enter the following:

```
:F 200 "This is more Text" 20,20 ;
```

## 12 The Graphics Editor

To add a text component identical to the last text component at a new location 100,100, it is only necessary to enter the following:

```
100,100 ;
```

The command, descriptor, layer number, font size, angle of rotation, and characters forming the text component are all remembered as the current command parameters. If you wish to enter a new descriptor or layer number for the current command, both the descriptor and optional layer number must be entered.

Entering a command name itself (i.e., ADD T) resets parameter values such as font size and angle of rotation to their default conditions. That command, complete with new parameter values, then becomes the current command.

### The HP EGS Grid System

There are three different grids: the system grid, the user grid, and the display grid.

- The system grid is all possible points.
- The user grid is a subset of the system grid which you can define as active for snapping and display.
- The display grid is a potential subset of the user-defined grid.

You can designate a subset of the user grid for display purposes, although all user grid points remain active for snapping whether they are displayed or not.

### The System Grid

When you create a drawing on a sheet of paper with pen or pencil, you have a finite drawing area bounded by the edges of the paper. This gives you an infinite number of locations where you may place your pen.

With HP EGS you also have a defined drawing area which is called the total drawing space. However, you are limited to a finite number of locations that you may specify within this drawing area. These locations are evenly distributed about the point 0,0, and each location is identified by a whole number in the system.

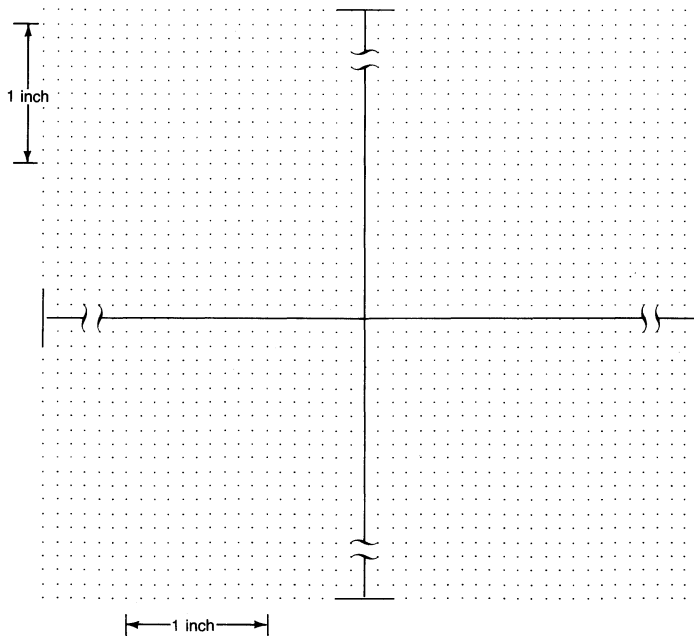
The array of these locations is called the system grid. There are 1 billion ( $10^9$ ) system grid points in both the + and - directions of the x and y axes. The UNITS command allows you to assign a physical distance to the system grid, thus setting the user units of the drawing. With the UNITS command you specify the number of system grid points that are equal to one physical unit such as an inch or a centimetre. Since the system units range is  $\pm 10^9$  in both the x and y axes, you set the maximum drawing size and the resolution at which locations can be accessed.

By using the UNITS default command you may assign the following units to the system grid:

micrometre	micro-inch
millimetre	mil
centimetre	inch
metre	foot
kilometre	yard
	mile

This allows you to work in units that make sense to your application.

For example, you can specify that the units used in your drawing are inches and that you want 10 system grid points equal to one inch in your drawing. Since you can enter xy locations only on grid points, this means that the smallest distance you can specify in your drawing is 0.1 inch. This is illustrated in the following drawing.



In the example above, the units of the system grid are set so that there is 0.1 inch between adjacent system grid points. When entering locations (i.e., an endpoint of a line that you are adding to the drawing), the location entered is by default snapped or shifted to the nearest user grid point.

## The User Grid

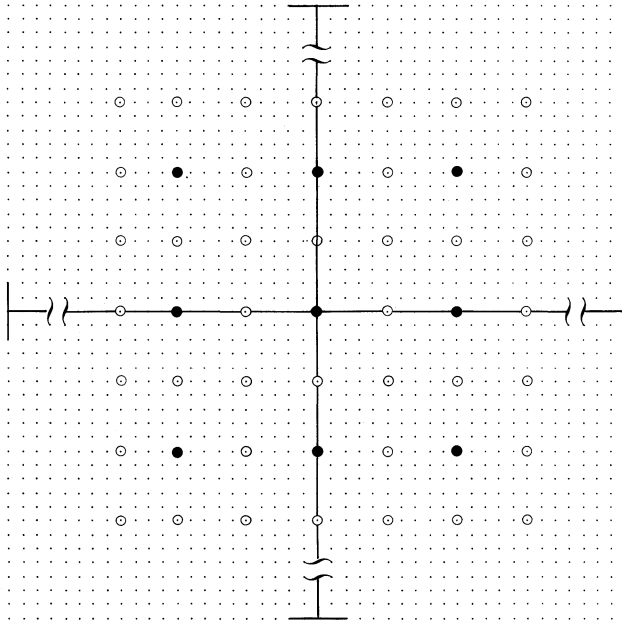
With the GRID command, you can define a subset of the system grid, called the user grid, so that only a subset of the system grid points are active.

For example, you could define a subset of the 0.1 inch system grid such that a user grid point exists every 0.5 inch. Therefore although a system grid point is defined every 0.1 inch, displaying only user grid points which are every 0.5 inch makes it much easier to add components to the drawing where the 0.1 inch resolution is not needed.

Snapping only applies to a user grid point. System grid points that are not active in the user grid cannot be snapped to.

## The Display Grid

The GRID command also allows you to limit the number of user grid points that are displayed in the drawing area and keep the drawing area uncluttered. This creates what is called the display grid. The system grid, the units assigned to the system grid, the user grid, and the display grid are all shown in the following drawing.



- System Grid Point
- User Grid Point
- Display Grid Point



Entering an xy location with the GRID command causes the grid to be shifted so that a grid point lies on the location specified. The grid is aligned with the location 0,0 if no alignment point is entered with the command. If you have a color system, you can specify the color of the grid.

The two values adjacent to the characters G: in the information display portion of the CRT provide information about the current user grid and display grid settings. For example, suppose G: .5,2 is displayed as the current user grid and display grid settings. The first value specifies the number of user units between user-defined grid points. A user grid point exists every 0.5 inches. The second value specifies the number of user grid points to be displayed. Every second user grid point is displayed. Each grid point displayed is 1 inch apart, since two user grid points times 0.5 inch per user grid point equals 1 inch.

Even though some user grid points are not displayed, they are still active for snapping.

### Selecting Drawing Points with Snapping Modes

When you create a drawing with HP EGS, you are always placing points on system grid points. At times you may want to be extremely precise regarding where a point is placed (i.e., exactly on a user grid point, at an intersection of two lines, etc.). To help you locate points, HP EGS lets you snap to different points using five “snapping modes.” These snapping modes are global commands that you may execute at any time, even in the middle of a primary or secondary command.

The five snapping modes are:

- :GRID            snapping to the nearest user grid point. This is the default mode.
- :PRIMITIVE    snapping to the nearest point on the component (primitive) closest to the cursor.
- :VERTEX        snapping to the vertex nearest to the cursor. For a line, polygon, or rectangle, the vertices are endpoints or corners. For a circle, the vertex is the center. For an arc, the vertices are its endpoints and the center. For markers, the vertex is the marker itself. The only vertex for notes, text, and instances is at the origin.
- :INTERSECT    snapping to an intersection on the component closest to the cursor or to the nearest vertex.
- :RAWPOINT     snapping to the nearest system point (or rawpoint) which gives the finest resolution possible, regardless of the current grid setting.

Once you have entered a snapping mode, it remains in effect until you change it. The current snapping mode is displayed at the bottom of the large viewport in the information display area (i.e., :GRID).

Since the system snaps to the nearest point which it determines using mathematical calculations, the placement of your stylus only needs to approximate the desired point. The area surrounding the cursor in which the system can sense the presence of a point is determined by the size of the cursor. You can set the cursor size with the CURSOR command. When primitive, intersect, or vertex snapping modes are turned on, and when no appropriate point is within the boundary rectangle defined by the cursor, the system defaults to grid snapping.

### **Dynamic Component Tracking**

Some commands have dynamic tracking of points for some components, if the points are entered from the graphics input device. In dynamic tracking, the system “rubber-bands” the component from a point already entered into the system to the point currently defined by the cursor. This allows you to adjust the location of the dynamically tracked point until it is located where you want it.

### **Evaluation of Mathematical Expressions**

You may want to include mathematical expressions in commands to have the system calculate the location of points and other values for you. This is allowable, providing that you use an expression which is legal in the EVALUATE command. Such expressions must be enclosed in parentheses.

The system evaluates the expression and substitutes the resulting value in the command. For example, a formula could be used to generate the x and y locations in an ADD or COPY command.

For more details, please refer to the EVALUATE command in the *HP EGS Syntax Reference*.

## HP EGS Drawings

This section discusses the basic geometric shapes that make up drawings and how drawings are stored. In addition, it discusses several topics you should consider in making drawings, such as assigning layers, nesting levels, logical levels, and associated text to components and using isometric grids.

### Drawings and Drawing Files

Drawings can exist temporarily in computer memory and permanently in a drawing file on your mass storage device. When a drawing is edited, the changes are made to the copy of the drawing in computer memory. By saving the edited drawing you update the copy of that drawing stored on your disc.

When a new drawing is created or an existing drawing edited or referenced, a copy of that drawing is placed in computer memory. The copy of the drawing remains in computer memory until you quit the Graphics Editor. Only one copy of the drawing resides in computer memory.

A library part is another name used to describe a drawing stored in a drawing file. Library parts are typically drawings which can be used over and over again, such as schematic symbols, nuts and bolts, or electrical parts. Special care should be given to the point of origin in a library part, as this allows for accurate placement of a library part in a drawing.

A library part may be defined in two different ways. First, it may be defined as a series of graphics components which you have combined to produce a drawing. These drawing instructions are stored in a drawing file ending with the suffix `_d`. Second, a library part may be defined as a “macro instance” which is simply a set of instructions telling how to create a type of drawing (i.e., a bolt). A macro instance differs from an ordinary instance of a drawing in that you must supply some values for its parameters (i.e., length and diameter) before the system can produce a specific drawing. Macro instances are text files ending with the suffix `_i` which contain the commands needed to make an instance of your drawing. When a macro instance is added to a drawing, the system prompts for each of the parameters.

Modifying a copy of a library part in computer memory does not automatically change the library part stored in a drawing file. However, if the modified part is stored back in its original drawing file, the definition of the library part is modified. Once a library part’s drawing file is modified, all drawings containing that library part will display the modified library part.

With the exception of `_d` drawing files, all other files created by the Graphics Editor can be accessed and edited by the Pascal Editor. Because drawing files are stored as binary files for better performance, they cannot be edited by the Pascal Editor.

## Drawing Components

HP EGS uses 12 basic components to create drawings. These components include 11 geometric figures (arc, circle, dimension, hatch, marker, line, note, oval, polygon, rectangle, and text) and instances, which are wrapped drawings made up of these geometric figures. All drawings are made up of these components or building blocks.

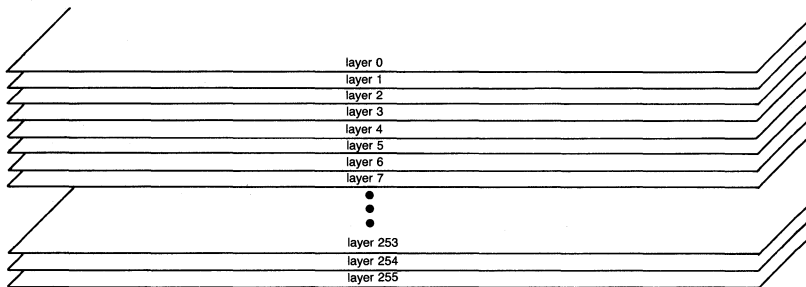
A descriptor is a character describing any of the components. The descriptors recognized by HP EGS are:

- |     |           |     |                |
|-----|-----------|-----|----------------|
| • A | arc       | • M | marker         |
| • C | circle    | • N | note           |
| • D | dimension | • O | oval (ellipse) |
| • H | hatch     | • P | polygon        |
| • I | instance  | • R | rectangle      |
| • L | line      | • T | text           |

For a complete description of all the available methods of adding components, please refer to the *HP EGS Syntax Reference*.

## Layers

The total drawing area can be thought of as a stack of 256 clear plastic sheets, called layers, upon which the drawing is created. Creating a drawing involves adding components to one of the layers numbered 1 through 255. Layer 0 is reserved for displaying instance boundaries. The layers can be displayed one at a time or in any combination with the SHOW command. Thus, you can call up an entire drawing but display only those portions that you want to see.



For example, an architect could create a floor plan that includes lighting fixtures, plumbing layout, and furniture placement. By entering the furniture placement on layer 1, the plumbing layout on layer 2, and the lighting fixtures on layer 3, the architect can view only the lighting, only the plumbing, only the furniture, or any of these in combination. From only one drawing on the system, any number of useful drawings can be displayed and plotted.

Before a layer is used, the following items may be defined in a process file for a personality:

- The layer display type: detail layer, symbolic layer, or interconnect layer.
- The line style: any of eight line types used for components drawn on this layer.
- The color with which components in the layer are displayed (required for color systems only).
- The plotter pen number for components in the layer.
- An eight-character layer label.
- The set of layers which are bound to the layer which is being defined.
- A layer number. While up to 255 layers can be defined at one time, you only need to define the layers you intend to use.

This is all done with the EQUATE command.

Different layers may be assigned different layer types: detail type, interconnect type, and symbolic type. When a display mode is set, only the subset of layers defined as that layer type are displayed. The layer types correspond to the HP EGS display modes of the CRT: detail mode, symbolic mode, and interconnect mode. The :S and :D options of the WINDOW command determine the display mode shown in the viewports.

- When the detail display mode is set, components on detail layers are displayed, along with components on interconnect layers.
- When the interconnect mode is set, only components on interconnect layers are displayed.
- When the symbolic display mode is set, components on symbolic layers are displayed, along with components on interconnect layers. In the symbolic mode, components with width are drawn as if they did not have width.

For example, the pads associated with nand gates could be on the detail layer, the traces connecting them could be on the interconnect layer, and the symbol for the nand gate on the symbolic layer. In the case of a plumbing layout, the detail layer might contain the actual parts making up a certain fixture, the interconnect layer might contain the pipes, and the symbolic layer might contain components representing the fixtures.

The layer display modes allow you to have two views of the same drawing in the viewports (i.e., one of layers defined as symbolic layers, and one of layers defined as detail layers). Components entered on interconnect layers are always displayed in both viewports. Using the WINDOW command, you can specify which mode to display.

Layer 0 must always be defined, and is used to display instance boundaries. Layers 1 through 100 and layer 255 are to be used in current and future versions of HP EGS.

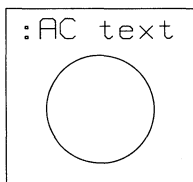
## Associated Text

Associated text is used to tie textual information to components and drawings. This concept is important for post-processors such as the HP EGS connection list utilities that extract textual information from graphical components. Note that although the term used here specifically refers to the text component, you may also use the note component in the same ways.

When a drawing is added as an instance to another drawing, the text that was originally associated with the instance drawing (:AD text) is copied and associated with an instance component (which makes a new piece of :AC text). This allows you to use the MODIFY command to change the text of that one instance, rather than changing the text of the permanently stored drawing. For example, you could add a 16-pin dip to a printed circuit board drawing and simply change the reference designator of that part from U1 to U1A. This change would apply only to one dip in the PC board drawing you are working on.

In contrast, when a drawing is added to another drawing, the text that was originally associated with components of a drawing (:AC text) remains associated to the individual components inside the instance of the drawing. No text is associated to the instance itself. Thus, the only way to change the text of the permanently stored drawing is to use the EDIT command to edit the instance, and then save the edited drawing. For example, you could change the stock number of a library part and have this change reflected in all the drawings that reference this part.

The following illustration shows how :AC and :AD text are transformed when a drawing containing them is added to another drawing:



Original  
drawing



Instance of  
original  
drawing

You may use the ADD T or ADD N commands to associate text and notes with components or drawings. The options for associating text include specifying:

- Whether or not the text is displayed.
- The tag value which classifies the associated text with other items with the same tag value. This allows you to ascribe meaning to associated text. The tag value may be any integer between  $-32768$  and  $+32767$ .
- Association of text to a particular component by pointing to that component.
- Association of text to a drawing currently being edited. This method is primarily used for library part drawings which will be added to other drawings. Thus, associated text is automatically added to instances of the library part when the part is added to a drawing.

For example, assume that two text components have been associated to the drawing `nand2`. If `nand2` is added as an instance to a drawing called `sch1`, the two text components associated to the drawing `nand2` will automatically be added to the drawing `sch1` and will be associated to the instance component `nand2`.

- The logical location to which the text points.
- The set of layers on which the logical location resides.

Whenever a component with associated text is selected and highlighted, the text is also highlighted (unless the text is undisplayed). If the text indicates a logical xy location, a line is drawn from the text origin to the logical location. The notes and text remain as components, meaning you can still move, rotate, scale, mirror, and change the text or note. If a component with associated text is moved, scaled, rotated, or mirrored, the orientation of the text is also changed.

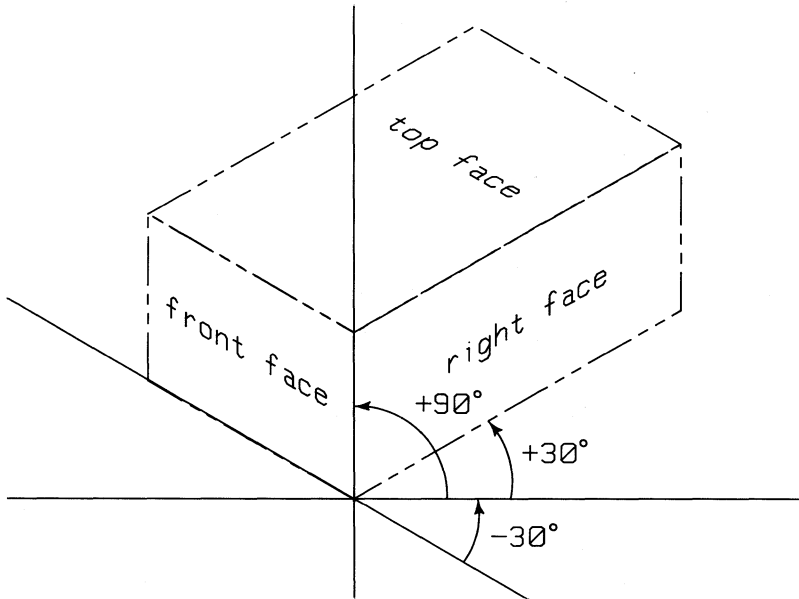
## Isometric Drawing

The HP EGS system allows you to make isometric drawings both quickly and conveniently. Although such drawings convey three-dimensional information, the system is strictly a two-dimensional drawing tool. The system neither uses depth information, nor provides the tools for making perspective drawings. Rather, the system provides tools that allow you to think in terms of one orthographic plane at a time, yet have the results automatically converted to and displayed as isometrics.

In HP EGS, an isometric plane is any plane in which the logical axes are not aligned with the orthogonal (right angle) screen axes. In isometric drawing, logical axes typically appear to be skewed (intersecting at angles other than  $90^\circ$ ) on the display screen, but they behave like they are orthogonal.

You can set the isometric mode with the GRID command or by using the screen menu macros for isometric drawing. Two isometric options of the GRID command are available:

- :I1 This option allows you to specify the angles from the positive screen x-axis to the two logical axes. For a normal 30-30 isometric drawing these angles are :
  - 30 and +90 for the front face.
  - +30 and +90 for the right face.
  - 30 and +30 for the top face.



- :I2 This option allows you to define the two axes by digitizing two points along each axis. The positive direction along the axis is the direction from the first point digitized towards the second point. This option is often used when the angles at which the axes intersect are not known.

To reset the grid to a normal setting, use the :I1 option with angles of 0 and 90 degrees. In any grid setting, the cursor changes to show the orientation of the axes. You may find it convenient to use a large cursor when drawing in the isometric mode. For more information on using the GRID and CURSOR commands, please refer to the *HP EGS Syntax Reference*.



### Isometric Conversions

The system makes the use of an isometric grid transparent to you by automatically converting the logical data you enter into the physical data drawn on the screen. This automatic conversion makes the construction of isometric drawings simple and quick. Note that the system saves absolute data (coordinate values based on the orthogonal screen axes). The isometric drawing tools provided by HP EGS are only a means of easily generating this absolute data.

The following summarizes the conversions that take place:

- |              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Points       | The definition of a point is two coordinates: x and y. The x coordinate is the distance of that point from the y axis measured parallel to the x axis. In an isometric grid, the logical axes are not parallel, but the same definition of coordinates applies. All values are assumed to be logical coordinates and are then automatically converted to absolute positions. The implication of this conversion is that the same physical point on the screen can have multiple coordinate values, depending on the coordinate system in which those values are measured.                                                                                  |
| Rectangles   | Rectangles are four-sided polygons specified by diagonally opposite corners whose sides parallel the logical axes. If you enter a rectangle with an isometric grid turned on, you will get the logically correct result. However, since the physical realization of what you draw represents a polygon rather than a rectangle, a polygon is saved in the drawing.                                                                                                                                                                                                                                                                                         |
| Circles/Arcs | When a logical circle or arc is drawn in an isometric mode, the result is an oval. Thus you can produce the correct result by using the ADD ARC and ADD CIRCLE commands to produce an oval with the appropriate orientation. Note that the oval you add is logically a circle in this isometric plane. This is important since in tangency calculations (as in ADD ARC, ADD CIRCLE, and ADD LINE), only lines and circles may be used. If an oval is logically a circle in the currently defined isometric plane, it may be manipulated just as if it were a circle. However, if you change planes there is no logical connection of the oval to a circle. |
| Lines        | When adding lines by specifying angles, the angle is measured relative to the logical x axis. Other methods of adding lines work like they would with a normal grid setting. Note that when adding a line tangent to an arc or circle, you may select an oval only if it is logically an arc or circle in the current isometric plane.                                                                                                                                                                                                                                                                                                                     |
| Polygons     | As polygons are specified by points, the addition of polygons is not affected by an isometric plane.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

Notes/Text	When notes and text are added in an isometric plane, their rotation is with respect to the logical x axis. Thus, if you add a note with no rotation, the note saved in your drawing will be rotated to align it with the logical x axis. Similarly, if you add text with no slant to an isometric drawing, a slant will be given to the characters to make them parallel with the logical y axis. If you add text with slant to an isometric drawing, that additional slant will be reflected in the drawing.
Markers	As a marker is specified by a point, the addition of markers is not affected by an isometric plane.
Ovals	The orientation of an oval is measured relative to the logical x axis.
Dimensions	Horizontal and vertical dimensions are drawn parallel to the logical axes. The textual information of a dimension (i.e., the value or tolerance) is transformed the same as notes and text. Note that the value calculated for a dimension is only accurate if the points defining the dimension all lie in the same logical plane. This means that if you consider the three-dimensional coordinates of the points defining the dimension, then one of the three coordinate values must be exactly the same for both points and that coordinate must match the current isometric grid setting.
Instances	Instances are not affected by an isometric grid. They are always brought in as an orthogonal view of the drawing.
Hatching	The angles are specified with respect to the logical axis.

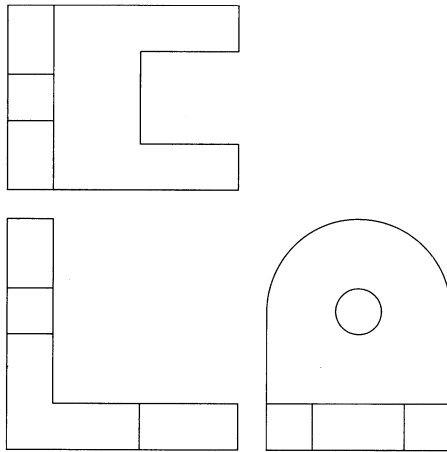
### **Converting Orthographic Views to Isometric Views**

As a result of the isometric conversions described in the previous section, you can easily convert an orthographic view of an object (typically top, right, or front view) into an isometric view using an archive file of a drawing. Note that this conversion is only one-directional, meaning you can convert an orthographic view into an isometric view, but you cannot convert an isometric view into an orthographic view.

An archive file is a text file containing the commands necessary to recreate a drawing. The bulk of an archive file is a series of ADD commands which add the various components that define the drawing. If these commands are input with an appropriate isometric grid turned on, the system automatically converts the drawing to an equivalent isometric view. Thus, by inputting modified archive files of three orthographic views of an object, the system can automatically perform most of the work necessary to produce an isometric view of that object for you.

The following procedure describes one way to perform this operation. It assumes you have prepared an archive file for a drawing that contains instances of each of the three orthographic views. An alternative procedure is to create three archive files, one for each of the three orthographic views.

1. Produce orthographic views of the objects you want to include in your isometric drawing. Do this using normal HP EGS commands but without using an isometric grid. Be sure each view is wrapped. For most drawings you probably will want to leave off any dimension values and add them later to the finished drawing. The following example shows three orthographic views of objects.



2. Archive the drawing of the three views that you produced in step 1. The name of the archive file saved by the system will contain the suffix `_a`.
3. Use the Pascal Editor to edit the archive file that you produced in step 2. Be sure to type in the suffix `_a`, (including the period), when you are asked for the name of the file you want to edit. (Note that steps 3 through 7 require you to use the Pascal Editor, so do not leave the Editor until instructed to do so.)

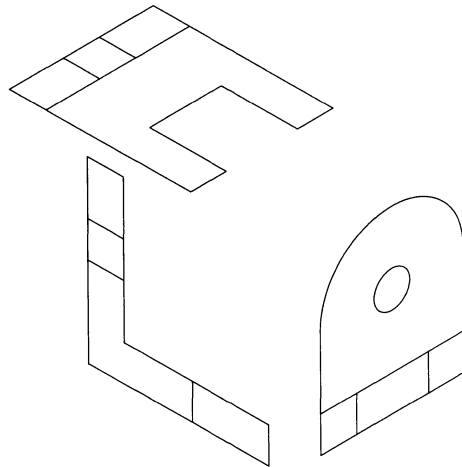
Find the command `$FILES`. Following this is a list of the files that make up this drawing. The first name in the list is the name of the main drawing, followed by the names of the various instances included in the drawing. (If you would like more information about the structure of archive files, please refer to the *HP EGS Syntax Reference*.)

4. The purpose of an archive file is to allow you to completely recreate a drawing from scratch. Thus the process of retrieving an archive file removes all the files listed in the `#FILES` statement, and recreates them with the information stored in the archive file itself. You will probably want to change the names of these drawings to unique names as you are going to produce an isometric view, and probably do not want to destroy the original orthographic views. In many cases you may not care to save separate isometric drawings of each instance, so you can convert them to unnamed instances.

Use the global find and replace function of the Pascal Editor to replace the names of the instances. Unnamed instances have temporary names such as `NONAMxxx` where `xxx` consists of three alphanumeric characters. If your archive file already contains unnamed instances (you can find out if it does using the Find Literal function to find the string `NONAM`), you should choose other names for the ones you are replacing. If you change any named instances into unnamed instances, delete the old file name from the line(s) of the `#FILES` command. Lastly, you will probably want to change the name of the main drawing to reflect that it is an isometric version of the original orthographic views.

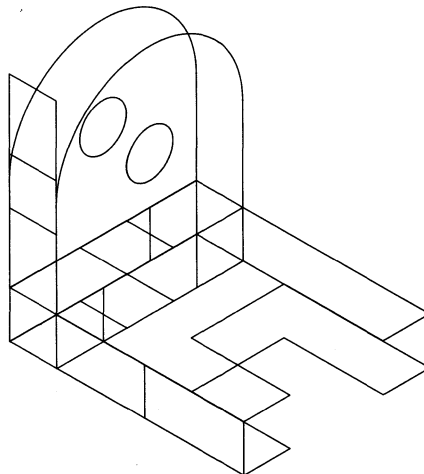
5. Once you have changed the names, find the beginning edit of each instance in the drawing. Each instance edit will begin with the sequence `EDIT xxxxxx`, where `xxxxxx` (up to 8 characters) is the name of the instance. Before the first `ADD` command, insert a `GRID` command to set the appropriate isometric grid. This will ensure that the logical values for the coordinates (which were absolute values in the original orthographic view) are automatically converted to new absolute values according to the specified grid. For example, if the file you are modifying is the orthographic view of the right side of the object, insert the command `GRID :I1 30,90;` in each instance edit. For an archive file containing all three views, place three distinct `GRID` commands in each of the edit blocks.
6. If you have changed the instance into an unnamed instance, change the `SAVE` command at the end of each sequence to an `EXIT` command.
7. Save this file as a text file. You may want to select a different file name since this is no longer an archive file of the original drawing. Use the Write option of the Pascal Editor to create a text file with a different name. Then exit the Editor.

8. While running the Graphics Editor, input the text file created in step 7. The system will automatically convert your orthographic views to the selected isometric views, as shown here:

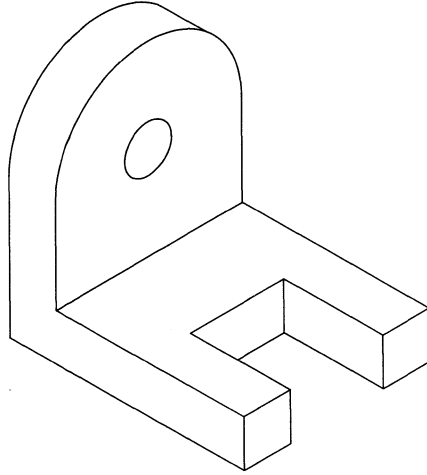


Note that you may input only one archive file at a time. If you have stored the parts of your drawing in separate archive files, rather than together in the same file, you will have to input each drawing separately and save it under a new drawing name. Then, starting from a clear screen, you can add each newly stored drawing to the screen (using the `ADD <instance>` command) until all the views appear.

9. Next move and copy the instances of the three faces of the object. This will create a wire-frame model of the object, as shown here:



10. Using the SMASH, DELETE, and DELETE :I commands, remove the hidden lines that you do not want in your final drawing. Add new lines as necessary to fill in the drawing. In general, lines not lying in one of the principal planes will need to be added to the drawing to complete it.



## Instances and Nesting Levels

The instance component is a component made up of other components. Although it contains multiple components, it is treated as a unit. This means that when you move an instance, all the components that make up the instance are moved, and when you delete an instance all the components that make up the instance are deleted. The individual components that make up the instance are not accessible to you, unless you destroy the instance with the SMASH command or look inside the instance with the WINDOW :N command.

Instances can be either named or unnamed. A named instance is stored in its own file and can be used in any drawing by simply adding a reference to it. An unnamed instance, on the other hand, may exist only in the drawing in which it was defined.

Instances offer two major advantages:

- They increase productivity by allowing you to use the same drawing over and over again. This is most commonly done by creating a library of symbols that can be used in many different drawings. A good example of this is the electrical schematic symbols supplied with the HP EGS Electrical Schematic Drawing personality.

- They reduce memory requirements. When an instance is added to a drawing, only a small amount of information is needed to draw that particular copy of the instance. This information includes:
  1. A reference to the instance. You may think of this as the name of the instance.
  2. The location of the instance in the drawing.
  3. The scale of the instance.
  4. The rotation of the instance.
  5. The mirror of the instance.

This information is all that is needed to draw the instance, regardless of the complexity of that instance. Thus when the first instance is added to a drawing, both the instance information and the drawing itself are loaded into memory. However, for each additional copy, only new instance information is stored.

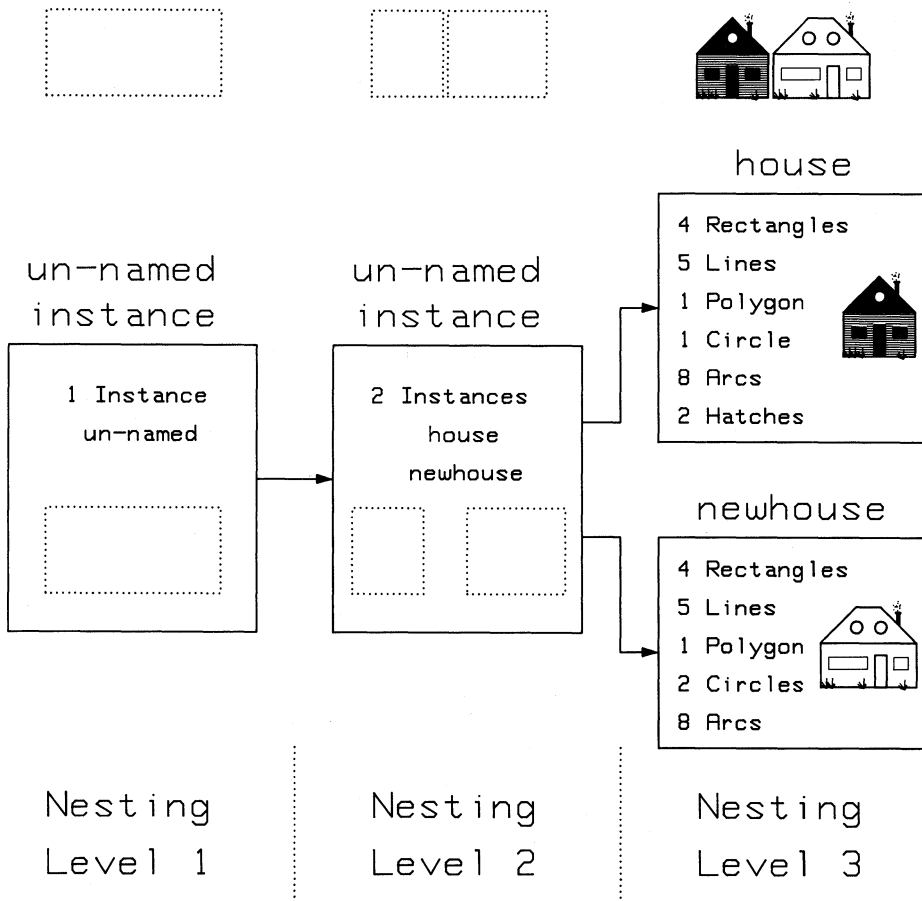
HP EGS allows you to define instances in two ways. The first method involves defining an instance by graphically creating the picture of that instance. Once the drawing is created and stored, you may use that drawing in any other drawing. To do so, you would simply use the ADD command and specify the name of the drawing (instance) you wish to add.

The second method of defining an instance is algorithmic. These instances, called “macro instances,” are defined by procedures for making a drawing based on its parameters. Several examples appear in the HP EGS Mechanical Drafting personality. One of them is a “generic” formula for producing a bolt. When you add an instance of the bolt, you must supply not only the information for a normal instance, but also the parameters that uniquely define this particular bolt. In this case, the parameters represent length and diameter.

Whenever a drawing containing instances is stored, only the instance information itself is stored. The definition of the “master” drawing (either graphic or algorithmic) is stored separately. Thus if you change the underlying definition of the instance by changing the file that defines it, every drawing that references that instance will automatically reflect those changes. This feature allows you to produce drawings in a top-down fashion. For example, an architect creating a floor plan might not yet have decided which size and model of desk to use as office furniture. By adding instances of a drawing named DESK, he can create a floor plan without worrying about the details. When he finally selects a desk, he needs only to create the drawing DESK; each drawing will automatically include it when it is redrawn.

HP EGS allows you to nest drawings within drawings and to preserve this drawing hierarchy. Instances can be nested to any level. This means an instance can contain instances, which contain additional instances, and so on. Thus you can create complex parts from a hierarchy of simpler parts. When HP EGS displays these parts, you may choose the level of detail you wish to see by using the WINDOW command and specifying the desired nesting level.

When you set the nesting level to 1, you see only simple components in your drawing. Each instance is represented by a dotted box outlining that instance. If you increase the nesting level to 2, you can see inside the first level of instances. This concept is illustrated in the following drawing from *Learning HP EGS*. At nesting level 1, you see one dotted box. At nesting level 2, you see two boxes. Only at nesting level 3 can you see the detail of both houses. This level represents the deepest nesting level of this drawing.





When you load a drawing that contains instances, the program automatically checks to see if the files defining those instances are already in memory. If not, the definitions are loaded. When you set the window nesting level to higher and higher values, you see successively more and more detail. At times, it is advantageous to suppress the detail in some of the picture while preserving it in other parts. A good example of this is a circuit diagram consisting of major blocks and interconnect information. If you are only interested in viewing the detail of some of the blocks, you can use the following technique:

1. Use the `LOAD <drawing>` command to load the blocks in which you do not wish to see the detail. You may load them to any nesting level. By using level 1, only the outline of the blocks will ever be visible.
2. Edit the circuit diagram that contains these blocks and the interconnect information. Those blocks not loaded in step 1 will be automatically loaded to their full depth and may be viewed by simply changing the nesting level. However, the blocks loaded in step 1 will only be visible to the level that you specified in step 1, regardless of the nesting level now chosen. This both increases the speed of redrawing and reduces memory requirements. Should you need to view one of the pre-loaded blocks in greater detail, you should use the `LOAD <drawing>` command again and specify a higher nesting level.

## Logical Levels

Another way to specify hierarchy within a drawing is with logical levels. This hierarchy is independent of the nesting level hierarchy just discussed. Unlike nesting levels, a logical level is *explicitly assigned* by you to an instance. This assignment is a number describing a conceptual hierarchy within a drawing. This feature is often useful when using the `GENERATE` and `ARCHIVE` commands to eliminate any unnecessary instances, and in the `LIST MATERIAL` command to smash unwanted instances or create hierarchical connection lists.

The logical hierarchy is an assignment of a logical level number to each instance. Higher numbers mean a higher position in the hierarchy, and lower numbers indicate more basic parts. Instances with a lower logical level number can be used to create other instances with the same or higher-level number, but cannot be used to create lower-level instances. Note that this order is the opposite of that used for nesting levels.

A default level number of 1 is assigned to all instances at the time of creation. This number is specified by the `LEVEL` command in a personality's process file. If an instance created by the use of the `WRAP` or `GROUP` command requires a different logical level number than the default of 1, the instance must be stored and then edited as an individual drawing; then the `LEVEL` command can be used to change the logical level number. All library parts supplied with the HP EGS system are assigned logical level 0.

The system checks instances specified as part of a drawing for a logical level equal to or less than that of the current drawing. The system then displays an error message if this condition is not met. An example of an error would be to use an instance that has a logical level number greater than 1 to create a new drawing (since a new drawing would have the default logical level of 1).

## HP EGS Commands

One convenient way to think about the HP EGS Graphics Editor commands is to group them according to the tasks they perform. The following summary of commands shows which commands perform various kinds of tasks. For more information on each command, please refer to the *HP EGS Syntax Reference*.

Default Commands	Information and Drawing Environment Commands	Drawing Creation and Editing Commands	Input and Output Commands	File and Volume Manipulation Commands	Commands to Define Other Commands and Menus
DECIMALS	AREA	ADD	ARCHIVE	CATALOG	DEFINE
DIM-DEC	CURSOR	BACKUP	CALIBRATE	CHANGE	MENU
DIM-DIR	DATUM	COPY	DUMP	FILECOPY	
DIM-EXTEND	DISPLAY	DELETE	ECHO	PACK	
DIM-FSIZE	DISTANCE	GATHER	EDIT	PREFIX	
DIM-RADIX	EQUATE	GROUP	EXIT	REMOVE	
DIM-SCALE	EVALUATE	MODIFY	GENERATE	SEARCH	
DIM-TFSIZE	HELP	MOVE	INPUT	VOLUMES	
DIM-TSLANT	IDENTIFY	NOP	LIST		
DIM-UNITS	LOCATOR	SMASH	LOAD		
FSIZE	MEMORY	STEP	PAGE		
GRID	PAUSE	STRETCH	PLOT		
INAME	POINT	UNDO	QUIT		
LEVEL	REVISION	WRAP	RETRIEVE		
LOCK	SHOW		SAVE		
RESOLUTION	TMENU				
TNAME	WINDOW				
TSLANT					
UNITS					

- **Default Commands** – These commands set the default values for the drawing environment. They are described in a personality's process file. Although these values are set for you when you enter an HP EGS personality, you may change them. Such a change is temporary if you make it while running the Graphics Editor, since this new default value applies only to the current editing session. If you are leaving the Graphics Editor and wish your changes to be permanent, you may save the modified process file with the SAVE command.
- **Information and Drawing Environment Commands** – These commands provide you with information about a drawing, and they allow you to manipulate your drawing environment while you work. The commands allow you to define the characteristics of the layers on which you draw, and specify default values for your drawing environment. In addition, they allow you to control what is displayed on the screen and to derive descriptive information (such as area and distance values) about drawings.
- **Drawing Creation and Editing Commands** – These commands allow you to create and modify drawings.
- **Input and Output Commands** – This group of commands is used to input files, store drawings in files, and output files to printers, plotters, and mass storage devices. HP EGS supports all ASCII and foreign characters supported by the Pascal operating system.
- **File and Volume Manipulation Commands** – This group of commands is used to manipulate HP EGS files and volumes. Several Pascal commands may also be used to manipulate files and volumes (see *Managing and Editing Files with HP EGS* for more information).
- **Commands to Define Other Commands and Menus** – These commands allow you to customize the system by creating new command macros and screen and tablet menus. With them you can tailor the system to suit your individual application.



# Customizing Your System

## Chapter 2: Customizing Your System

Introduction. . . . .	35
Naming Conventions. . . . .	36
Defining the Personality. . . . .	38
Space Planning Example. . . . .	39
Making a Volume or Directory . . . . .	41
Space Planning Example. . . . .	41
Building a Control File . . . . .	42
Control File Structure . . . . .	42
Using the INPUT Command in Control Files . . . . .	46
User-written Applications. . . . .	46
Space Planning Example. . . . .	47
Building a Start File . . . . .	48
Space Planning Example. . . . .	48
Building Volume Search Tables . . . . .	50
Space Planning Example. . . . .	51
Building or Modifying a Process File. . . . .	52
Defining Layers . . . . .	52
Specifying Default Values . . . . .	53
Space Planning Example. . . . .	54
Building or Modifying a Screen Menu . . . . .	57
Screen Menu File Structure. . . . .	58
How The System Interprets Screen Menu Items. . . . .	59
Space Planning Example. . . . .	60
Building or Modifying a Macro File. . . . .	62
Defining a Macro . . . . .	62
Short Macros vs. Long Macros . . . . .	63
How the System Interprets Information. . . . .	64
Macro Substitutes for Commands . . . . .	65
Temporary Macros. . . . .	66
Macros to Save Values and Set Modes . . . . .	67
Macros to Modify Existing Information. . . . .	68
System-defined Macro Constants. . . . .	68
Conditional Macros . . . . .	75
Space Planning Example. . . . .	76

Creating Library Parts . . . . .	77
Special Drawing Considerations . . . . .	77
Macro Instance Library Parts . . . . .	78
Space Planning Example. . . . .	83
Building or Modifying a Tablet Menu . . . . .	85
Drawing a Tablet Menu. . . . .	87
Plotting a Tablet Menu . . . . .	89
Activating a Tablet Menu. . . . .	90
How the System Interprets Menu Items. . . . .	91
Space Planning Example. . . . .	91
Modifying the Message Files . . . . .	93
Structure of a Message File . . . . .	93
Runtime Parameters . . . . .	93
Editing and Indexing a Message File . . . . .	94
Examples of Message Files . . . . .	95
Help Messages for Personalities . . . . .	96
Creating New Stroke Tables . . . . .	97
The Stroke Table Format . . . . .	97
Example . . . . .	98
Troubleshooting. . . . .	99
Putting the Personality Together. . . . .	100
Space Planning Example . . . . .	100

# Customizing Your System

Chapter

2

## Introduction

This chapter discusses how to get the most out of your HP EGS system by tailoring it to meet your needs. The HP EGS system and its Graphics Editor may be completely customized, allowing you to perform your job more effectively and efficiently.

Before reading this chapter, you should have a working knowledge of the General Drawing personality presented in the first lessons of *Learning HP EGS*, and you should feel comfortable working with the Pascal Editor and Filer. If you need more information on using the Pascal Workstation, please refer to *Managing and Editing Files with HP EGS*.

The Graphics Editor personalities supplied with your HP EGS system – General Drawing, Mechanical Drafting, Electrical Schematic Drawing, and Printed Circuit Board Layout – were created using the steps listed below. Similarly, you can create other complete personalities or modify any part of an existing personality to suit your needs.

Customizing steps include:

1. Defining the personality and its library parts.
2. Making a volume or directory for the new personality.
3. Building a control file.
4. Building a start file.
5. Building volume search tables.
6. Building or modifying a process file.
7. Building or modifying a screen menu.
8. Building or modifying a macro file.
9. Creating library parts.
10. Building or modifying a tablet menu.
11. Modifying the message files.
12. Creating new stroke tables for notes and text.
13. Putting the personality together.

One good way for you to familiarize yourself with this process is to look at the contents of the files that define the personalities supplied with your system. These files are listed in Chapter 3: *Files, Volumes and Mass Storage*.

To help you understand customizing your system, this chapter demonstrates the steps involved in building a new personality designed to do space planning. This Space Planning personality is included with your HP EGS system on a disc named EWSP:. Each of the following sections contains a discussion of relevant concepts and procedures needed to build a new personality. Additionally, one or more examples relating to the Space Planning personality are included. You may view these examples by simply inserting the EWSP: disc in one of your computer's disc drives, and executing `HP EGS`. The Space Planning personality will then appear as one of your menu choices.

### Naming Conventions

In order to distinguish between different parts of a personality, HP EGS has naming conventions for data and information stored on your disc. To create new files for any personality, you must use the following file name suffixes:

<code>_pr,TEXT</code>	Graphics Editor process file
<code>_mc,TEXT</code>	Graphics Editor macro file
<code>_mn,TEXT</code>	Graphics Editor screen menu file
<code>_tm,TEXT</code>	Graphics Editor tablet menu file
<code>_st,TEXT</code>	Graphics Editor stroke table
<code>_cn,ASC</code>	Control file that defines menus for the system manager
<code>_ms,ASC</code>	Message file
<code>_mi,ASC</code>	Index to a message file (for quick loading)



<code>_vs,TEXT</code>	Volume search table
<code>strt,TEXT</code>	Graphics Editor start file for a personality (this suffix is optional)
<code>_d</code>	Graphics Editor drawing file
<code>_i</code>	Graphics Editor macro instance
<code>_c</code>	Connection list output (from Connection List utility)
<code>_e</code>	List of errors from the connection listing
<code>_m</code>	Graphics Editor material list output (from LIST MAT command)
<code>_pf</code>	Parts file for material list post-processors
<code>_a</code>	Graphics Editor ARCHIVE output
<code>_g</code>	Graphics Editor GENERATE output
<code>_ec,TEXT</code>	Graphics Editor ECHO file output
<code>_r</code>	Rat's nest output (from Generate Rat's Nest utility)

---

#### SRM vs. Local Disc Notation

Depending on the type of hardware you are using with your HP EGS software, you may be using local disc or Shared Resource Manager (SRM) notation to type in information. Throughout this chapter the examples show local disc notation. Be aware that if you try to duplicate any of these examples on your SRM system, you may need to alter the syntax.

The primary difference between local disc and SRM notation is the symbol used as a delimiter. Use a slash mark / for an SRM, and a colon : for a local disc. Please refer to *Managing and Editing Files with HP EGS* for more information on the differences between SRM and local disc notation.

---

## Defining the Personality

It's a good idea to think about all the things you want your new personality to do, both now and in the future, before you proceed with the customizing steps listed earlier. This preplanning can help you better design the commands and library parts you will need. You can avoid some problems that may later require valuable time to correct. Think about these topics:

**Purpose:** What is the purpose of your new personality, or the purpose of the changes you plan to make? You need a clear idea of what you want your personality to do before starting to create or manipulate files. Think about what library parts you will need to do the job, how you expect to manipulate your drawings, what you want your final drawings to look like, if you need material and connection listings, and so on.

**The User:** Who will be using this personality? How did that person previously create drawings? Try to make the user comfortable with your new personality. For example, a draftsman may prefer working from a graphics tablet and drawing with a pen, while someone else may prefer working entirely at the keyboard. If your user is an inexperienced computer operator, you may want to minimize the number of commands the user needs to know to do the job, or perhaps modify the messages that appear on the screen.

**The Job:** What kinds of drawings will you create and how will you manipulate them? The answers to these questions can help you decide what commands and macros are appropriate for your personality. You may want to design screen and tablet menus to accommodate frequently used commands and command options. And you may want to specify drawing units, line styles and colors for various layers, pen colors for plotted drawings, and other drawing standards in your process file.

**Library Parts:** How many parts will you use in your drawings? How often will you use each part? Will you be making and saving parts as you edit bigger drawings? Do you need to retrieve drawings from other volumes within HP EGS? Do you plan to compare or merge connection listings among drawings? The answers to these questions will help you decide what items to place on the various layers, where to place physical and logical ports on parts, tag values to assign to parts, how to store parts on different volumes, and how to design screen and tablet menus for easy use.

**Volumes and Files:** What HP EGS volumes and files do you need to create or change? Answering all of the above questions and looking at the existing personalities' files will help you decide which files to be concerned about. Make a list of these files so you can keep track of the work you do.

As you design and build your personality, remember that this process is iterative, meaning that you can update and refine your work as you go. You need not create all the parts or define all the macros at once – some of this work can wait until later. In fact, you may want to try using each feature of your personality as you go, testing a command, adding a library part, or calling up a file to make sure everything is working correctly. This trial-and-error method can help you detect flaws in your personality early and save you from having to make extensive corrections later.

Also remember to design your personality for maximum future flexibility. You may want to add commands, parts, and other features later, or you may change your hardware configuration. These changes could translate into a great deal of work if you do not anticipate and plan for them.

## Space Planning Example

This section illustrates each customizing step by showing you an example: the Space Planning personality. Let's look at one possible way to define this personality:

### Purpose:

- To make floor plans of a company work area. This process involves moving office furniture, phones, and wall dividers.
- To keep track of the amount of floor space used.

### The User:

- The manager of a publications department. This person is an inexperienced computer user who has some knowledge of drafting.

### The Job:

- Using screen and tablet menus to create and change individual drawings and entire floor plans. We could use the following drawing commands: ADD, COPY, DELETE, MODIFY, MOVE, UNDO, SMASH, and WRAP. We could also use the default and information commands GRID, WINDOW, and IDENTIFY; write new OPEN\* and CLOSE\* macros to display furniture in opened and closed positions; and add frequently used options for commands to menus. To help us locate points in the drawings, we could set the snapping mode to :GRID, :VERTEX, :INTERSECT, :PRIMITIVE, or :RAWPOINT.
- Calculating square footage requirements. To do this we could use the information commands AREA and DISTANCE.
- Plotting furniture drawings. To do this we could use the output commands PLOT and DUMP.

### Library Parts:

- Desks, drawers opened and closed
- Tables
- Chairs
- File cabinets, opened and closed
- Desk and secretaries' phones
- Wall dividers of varying size

### Volumes and Files:

- |                                       |                     |
|---------------------------------------|---------------------|
| ● Volume Name                         | EWSP:               |
| ● Control File                        | EWSP:sp_cn.ASC      |
| ● Manager Volume Search Table         | EWSYS:hpegs_vs.TEXT |
| ● Start File                          | EWSP:spstrt.TEXT    |
| ● Graphics Editor Volume Search Table | EWSP:sp_vs.TEXT     |
| ● Graphics Editor Process File        | EWSP:sp_pr.TEXT     |
| ● Graphics Editor Macro File          | EWSP:sp_mc.TEXT     |
| ● Graphics Editor Screen Menu File    | EWSP:sp_mn.TEXT     |
| ● Graphics Editor Tablet Menu File    | EWSP:sp_tm.TEXT     |
| ● Graphics Editor Tablet Menu Drawing | EWSP:sptmenu_d      |

## Making a Volume or Directory

One good way to organize the files for your new personality is to store them in a volume (for a local disc) or directory (for an SRM). Keeping them in such a central location avoids any confusion with the files from other personalities.

Make your volume or directory using the Pascal Filer (pressing   for Make Directory on an SRM and  for Change Name of a Volume on a local disc). You may use up to 6 characters (upper or lowercase) to name a volume on a local disc, and up to 15 characters (upper or lowercase) to name a directory on an SRM. Although any name may be used, it is often a good idea to distinguish HP EGS volumes and directories from other storage areas – notice how all the volumes or directories supplied with your system begin with the letters EW, which stand for Editor Workstation.

Once you have created a storage place, you may start building and modifying files and storing them in this new location. Make sure your file names conform to the naming conventions for HP EGS mentioned earlier.

### Space Planning Example

The volume used to store files for the Space Planning personality is:

EWSP:

You will find this disc included with your HP EGS system.

## Building a Control File

When you first enter HP EGS, you may select from various options on different menus. This program you have entered is called the Manager. The information contained in these menus comes from special files called “control files.” Each personality supplied with the HP EGS system has a separate control file. In order for you to include your personality in the Manager menus, you must also create one of these control files.

A control file is a special message file contained within your personality’s volume or directory. It must have the suffix `_cn,ASC`. This suffix allows the Manager to recognize it as a control file. When the Manager is first executed, it searches all volumes and directories described in the volume search table `EMSYS:HP EGS_VS,TEXT` for control files. Then the Manager reads and builds the menus you see on the screen from these files.

### Control File Structure

Each control file contains information that describes one or more menus. This information must be formatted and structured in a special way. The following example shows the required structure:

```

menu_name  ^D^TEXT
           <action>
           extra1  (if necessary) } (repeated as many times as necessary)
           extra2  (if necessary) }

```

```

menu_name2 ^D^TEXT
           <action>
           extra1  (if necessary) } (repeated as many times as necessary)
           extra2  (if necessary) }

```

The items in this file specify the following:

- `menu_name` The menu name can be a maximum of 20 characters long. This name describes the menu being defined. There is only one reserved menu name in HP EGS, and that is "main". Any group of menu items named "main" is included in the first menu which appears when you enter HP EGS. Menu names of the same name may appear in a separate control file. The Manager automatically combines the menus described in different control files to form one comprehensive menu.
- `^D^` These special characters act as a delimiter between the menu name and the body of the menu being defined. They must always be in the control file.
- `TEXT` This is the text which will appear in the Manager menu. This text may be a maximum of 40 characters long.
- `<action>` The action item instructs the Manager what to do if you select a given menu item. The list of acceptable action items includes:

<code>&lt;execute_assembler&gt;</code>	executes the Pascal Assembler
<code>&lt;execute_compiler&gt;</code>	executes the Pascal Compiler
<code>&lt;execute_editor&gt;</code>	executes the Pascal Editor
<code>&lt;execute_filer&gt;</code>	executes the Pascal Filer
<code>&lt;execute_librarian&gt;</code>	executes the Pascal Librarian
<code>&lt;menu&gt;</code>	transfers to another menu
<code>&lt;execute&gt;</code>	executes any program you want
<code>&lt;quit&gt;</code>	quits and stops the Manager
<code>&lt;null&gt;</code>	no action is taken
<code>&lt;reboot&gt;</code>	powers-up the system
<code>&lt;reread&gt;</code>	looks in drives for flexible discs

`extra1` This line of information is required for all of the action items except `<quit>` and `<null>`. If the action you want is `<execute_assembler>`, `<execute_compiler>`, `<execute_editor>`, `<execute_filer>`, or `<execute_librarian>`, the third line is any text (maximum of 8 characters less than your screen width) you want to pass to a Pascal operating system program via the type-ahead buffer. (To find out the names of the Pascal system programs, use the What command from the main Pascal Command Line). To indicate carriage returns in the type-ahead text, use `<cr>`. If you leave the type-ahead line blank, no text will be passed.

If the action you want is `<execute>`, the `extra1` line is the full name of the program you want to run.

If the action you want is `<menu>`, the `extra1` line is the menu name you want to transfer control to.

## 44 Customizing Your System

`extra2` This line is only allowed when the action is `<execute>`. This line is any text (maximum of 8 characters less than your screen width) you want to pass to a Pascal program via the type-ahead buffer. To put carriage returns in the type-ahead text, use `<cr>`. If you leave the type-ahead line blank, no text will be passed. If you leave the program name line (`extra1`) blank, the Manager will terminate and the type-ahead text will be placed in the buffer.

Let's look at the control file `EWSYS:hpe$g_s_cn,ASC:` as an example.

```
main      ^D^System Utilities
          <menu>
          utils
          General Drawing Editor
          <execute>
          gedit.CODE
          INP gdstrt,TEXT; <cr>
utils     ^D^RETURN To Main Menu
          <menu>
          main
          Media Initialization
          <execute>
          *MEDIAEGS

          CS80 Tape Backup
          <execute>
          *TAPEEGS

          <null>
          Editor
          <execute_editor>

          Filer
          <execute_filer>

          Pascal Command Line
          <quit>

          <null>
          Break-up A Large File
          <execute>
          filbrk.CODE
          filbm<cr>
          Re-create A Large File
          <execute>
          filmnd.CODE
          filbm<cr>
```



```

Mag/Paper Tape Utility
<execute>
mputil.CODE
mputil<cr>tpdrv<cr>
Character Code Converter
<execute>
codeconv.CODE
cconv<cr>
Message File Utility
<execute>
testmfile.CODE

```

```

<null>
Install Codeword
<execute>
EWCODE:codeword.CODE
EWCODE:cdwd.ms,ASC<cr>
Re-boot System
<reboot>
Reconfigure System
<re read>

```

Here is a summary of the possible menu descriptions and action lines for control files:

Action Desired	Format for the Control File
Execute a Pascal routine	Menu text <execute_????> Information for the type-ahead buffer (where ???? = assembler, compiler, editor, filer, or librarian)
Execute a program	Menu text <execute> Name of the program to be executed Information for the type-ahead buffer
Call a menu	Menu text <menu> Name of the menu to transfer to
Quit the Manager	Menu text <quit>
Do nothing	Menu text <null>

Note that control files are sensitive to the number of lines per entry, and that blank spaces at the end of a file are not permitted.

For all control files:

- Extra blanks inserted at the beginning or end of any line will be removed.
- Any line starting with a { and ending with a } will be treated as a comment and is ignored.

### Using the INPUT Command in Control Files

When you create a control file, you may want to add type-ahead information so that the system performs actions automatically when a program is executed. If the program you are executing is the HP EGS Graphics Editor, you can enter commands from a file rather than the keyboard. You can use the INPUT command of the Graphics Editor to enter more information than the type-ahead buffer allows. Although your type-ahead buffer still may be a maximum of 8 characters less than your screen width, the file you INPUT to the Graphics Editor may be any size.

Common operations performed with the INPUT command include:

- Loading default menu, macro, volume search, and process files.
- Loading a particular drawing and setting the appropriate grid. This is particularly useful if you are working on a single drawing for a long time.
- Loading frequently used commands to increase performance.

The control files supplied with the HP EGS system normally use the INPUT command to load commonly used commands and menu, macro, and process files for a given personality. For information on other ways to use the INPUT command, please refer to the *HP EGS Syntax Reference*.

### User-written Applications

Your application programs may be integrated with the applications offered by the system. They may be selected from the Manager menu, just like any other application, if they are described in a control file.

Application programs are often used to:

- Post-process drawing data (convert it from its present form to another form).
- Perform analysis or other special tasks.
- Compile and print reports.

An important point to remember for any application program placed inside of HP EGS is that all external libraries must be linked to the program, or the program will not run. The Manager is linked with the following Pascal operating system files: LIBRARY, GRAPHICS, INTERFACE, IO, and SEGMENTER. In addition, the program being loaded by the Manager cannot use more than 32000 bytes of global space.

### Space Planning Example

The following example shows you how you might add a choice for the Space Planning personality to the Manager's main menu. You may look at this control file on the disc EWSP: which was shipped with your system.

The control file for the Space Planning personality is:

```
EWSP:SP_CN.ASC
```

It contains the following information:

```
main      ^D^Space Planning Editor
          <execute>
          $edit.CODE
          inp EWSP:SPstrt.TEXT;<cr>
```

Notice in this example that there is only one menu entry in the control file. This menu entry is to be part of the main menu of the Manager. If selected this menu item will execute the program `$edit.CODE`, which is the HP EGS Graphics Editor. Once `$edit.CODE` has been executed, the system will input the file `SPstrt.TEXT` from the volume `EWSP:`.

## Building a Start File

In the previous example, a start file was input when the Graphics Editor was executed. Although a start file is not required to run the Graphics Editor, it does provide a convenient way to load all the files necessary to work with a given personality. The files normally needed to use a personality include the macro, screen menu, tablet menu, volume search, and process files. As shown in the menus for the HP EGS personalities, you need not individually load these files every time you choose your personality from the Manager's menu – simply use the INPUT command to call up your personality's start file.

Any command listed in the *HP EGS Syntax Reference* may be used in a start file. The most commonly used commands are LOAD and PREFIX.

When you first enter an HP EGS personality, only a small number of Graphics Editor commands (listed in the start file) is loaded into current memory. If you select a command that is not loaded, the system searches for it on your mass storage device. Thus, if you want to get a quick response from the system when you select a command, you can place LOAD COMMAND statements in the start file to bring frequently used Graphics Editor commands into current memory. In contrast, if you find that you are running out of memory space while working with HP EGS, you may want to remove some loaded commands from your start file and re-enter HP EGS.

To create a start file, use the Pascal Editor to enter your information. Save this file under a name that has the letters denoting your personality and the suffix `start.TEXT`. (Note that this suffix is not required in order for the system to recognize this file; it simply conforms to the naming conventions followed for other HP EGS files.)

### Space Planning Example

The start file for the example Space Planning personality is:

```
EWSP:SPstart.TEXT
```

This file contains the following information:

```
loa com load;
loa com define;
loa com menu;
loa com window;
loa com add;
loa com delete;
loa com modify;
loa com move;
loa com copy;
loa com wrap;
pre EWSP:;
loa mac sp;
loa pro sp;
loa sea sp;
loa men sp;
loa tmen sp;
```

Given this information the system loads the commands LOAD, DEFINE, MENU, WINDOW, ADD, DELETE, MODIFY, MOVE, COPY, and WRAP. In addition, it prefixes to the volume EWSP: and loads macro, process, volume search, screen menu, and tablet menu files. Although commands and file names are shortened here for convenience, you could type in entire commands and file names and get the same result. The syntax abbreviations used in this example are explained in more detail in the *HP EGS Syntax Reference*.

It is important to remember that start files will only work for the HP EGS Graphics Editor. This concept will not work for other user-written programs.

## Building Volume Search Tables

Before you can work with any personality, HP EGS must be able to find the personality's control file and the other files you want to manipulate. Thus, you must tell the system where to look for these things by specifying the appropriate volumes in volume search tables.

Two volume search tables are of interest: one associated with the Manager's volume and one associated with the Graphics Editor's personality volume. The Manager's volume search table `EWSYS:hpegs_vs.TEXT` tells the Manager where to look for control files displaying menu choices. The personality's volume search table `EWXX:xx_vs.TEXT` tells the system where to look for drawings and other files required by the Graphics Editor.

You will need to change the Manager's volume search table if you have added a new volume or directory to HP EGS. Use the Pascal Editor to add the new information to `EWSYS:hpegs_vs.TEXT`.

You will need to add or modify a personality's volume search table if you want the system to search for files in volumes other than the one to which you are prefixed or other than those listed in an existing volume search table.

The volume search table is important to the Graphics Editor and your personality. When you tell the Graphics Editor to load a drawing or any file, one of two things will happen. If you have specified the exact volume or directory name, the Graphics Editor will search only where you have told it to look. But, if you have not specified a volume or directory name, the system will search first in the volume to which you are prefixed and then in top-down order through the volumes listed in the volume search table. If a drawing or file name you have requested appears in more than one volume, the system finds the first occurrence.

To create or change the volume search table for the HP EGS Manager, use the Pascal Editor. To create or change a personality's volume search table, use either the Graphics Editor commands `SEARCH`, `LOAD SEARCH`, and `SAVE SEARCH`, or the Pascal Editor.

There is no practical limit to the number of volumes and directories which may be in a volume search table. However, the maximum number of characters is limited to 120 per line.

## Space Planning Example

To enter the new volume EWSP: to the Manager's volume search table, we must edit the file named:

```
EWSYS:hpe@s_vs.TEXT
```

Once edited, this file should contain the following information:

```
EWSYS:  
EWME:  
EWEE:  
EWRC:  
EWOPT:  
EWSP:
```

Of course, we could delete the other volumes from this table if we wanted, but this would mean that the Manager would not find the other personalities.

We must also create the new personality's volume search file for the Graphics Editor. It is named

```
EWSP:sp_vs.TEXT
```

and includes the following information in it:

```
EWSP:
```

This volume search file tells the system to search for files in the volume EWSP: only. If we wanted to work with drawings and text contained in other volumes, we could list more volume names in this file.

## Building or Modifying a Process File

A process file is a text file that defines the underlying drawing environment for the Graphics Editor and your personality. It specifies the layers on which you can draw and the various default values that characterize your drawings. Process files contain various default commands that define your drawing specifications, and EQUATE commands that define layer characteristics. Process files always end with an EXIT command.

You can create and modify process files using either the Graphics Editor or the Pascal Editor. To create a new file, it is easiest to use the Pascal Filer and Editor to copy an existing file and edit the copy. To change a process file while you are working on a drawing, it is easiest to use the Graphics Editor. Note that changes made from the Graphics Editor will apply only to the current editing session, unless you save the edited process file before leaving the Graphics Editor. The SAVE command also automatically formats your process file for you.

Although you may create and modify many different process files, you may load (using the LOAD PROCESS command) only one into the Graphics Editor at any one time.

### Defining Layers

Creating a drawing with HP EGS involves adding components to one of 255 layers. These layers must be defined in a process file before you can use them for drawing. To define each layer you plan to use, use the EQUATE command to specify the following items:

- the layer number (1 through 255, 0 is always reserved for system use)
- the layer label (up to 8 characters, name must be unique, i.e., not used to name other layers, commands, macros, or library parts)
- the layer type (detail, symbolic, or interconnect; you can turn these layer types on or off as a group using the WINDOW command)
- the line type for components drawn on this layer (solid, dashed, etc.)
- the color for components shown on this layer
- the plotter pen number for components plotted from this layer (0 through 8, with 0 meaning that components will not be plotted; pen 0 is often used to conceal items you don't want in your final drawing, such as construction lines, open desk drawers, etc. that were used to aid in making a drawing)
- any layers which are bound to the layer being defined (binding allows you to use the SHOW command to turn a layer on or off in conjunction with another layer)

Although you may define all 255 layers at one time, you need only define the layers you wish to use. HP EGS requires that layer 0 always be reserved for displaying instances. Although layers 1 through 100 and layer 255 are to be used in current and future versions of HP EGS, they are also available for your use.



For more information on the concept of layers, please refer to Chapter 1: *The Graphics Editor* in this manual. The complete explanation of how to use the EQUATE command appears in the *HP EGS Syntax Reference*.

## Specifying Default Values

In addition to defining layers, a process file can specify certain drawing environment defaults (i.e., values the system assigns if you do not specify other drawing environment values). The default commands that may appear in a process file are:

**UNITS** – specifies user units and the number of system points per unit (system points are the points on which you may draw).

**LOCK** – specifies the lock angle for lines and rotations.

**GRID** – specifies the user and display grids, which may include all or only some of the system points.

**DECIMALS** – specifies the number of decimal places used for numerical outputs.

**FSIZE** – specifies the default font size for all notes and text.

**TSLANT** – specifies the default text slant angle for all notes and text.

**RESOLUTION** – specifies the default resolution for arcs and circles.

**DIM\_FSIZE** – specifies the dimension line's font size.

**DIM\_TFSIZE** – specifies the dimension line's tolerance font size.

**DIM\_TSLANT** – specifies the dimension line's slant angle.

**DIM\_DEC** – specifies the number of decimal places in dimensions.

**DIM\_SCALE** – specifies a scaling factor for dimension values.

**DIM\_RADIX** – specifies the radix symbol (comma or period) for dimension values.

**DIM\_EXTEND** – specifies the distance from an object to the dimension extension lines.

**DIM\_DIR** – specifies the default direction (horizontal or vertical) for dimension values.

**LEVEL** – specifies the default logical level for components.

**DIM\_UNITS** – specifies the default units for line, radius, and diameter dimensions.

**INAME** – specifies the translation for INAMEs (identify names) from EGS/45 archive files.

**TNAME** – specifies the translation for TNAMEs (trace names) from EGS/45 archive files.

For more information on using each of these default commands, please refer to the *HP EGS Syntax Reference*.

## Space Planning Example

The process file for the example Space Planning personality is:

```
EWSP:SP_P.P,TEXT
```

This file specifies some possible default values for the space planning drawing environment and defines each layer used by the personality:

```
UNITS FOOT,120;
LOCK 0.0000;
GRID 0.250,2 0.000,0.000;
DECIMALS 3;
FSIZE 1.000;
TSLANT 0.0000;
RESOLUTION 0.0000;
DIM_FSIZE 1.000;
DIM_TFSIZE 1.000;
DIM_TSLANT 0.0000;
DIM_DEC 3;
DIM_SCALE 1.000;
DIM_RADIX ",";
DIM_EXTEND 1.000;
DIM_DIR HORIZONTAL;
LEVEL 1;
DIM_UNITS FEET/INCHES;
INAME :L1 :F1,000 :T1001;
TNAME :L1 :F1,000 :T1004;
##
##          Layer Line
## Label   Type  Type  Color  Pen   Layer
EQU INSTBOUN :D    :LB   :C1   :P1   0;
EQU DESKS    :D    :L1   :C2   :P1   1;
EQU CHAIRS   :D    :L1   :C3   :P2   2;
EQU TABLES  :D    :L1   :C4   :P3   3;
EQU CABINETS :D    :L1   :C5   :P4   4;
EQU WALLS    :D    :L1   :C6   :P5   5;
EQU PHONES   :D    :L1   :C7   :P6   6;
EQU OPENFURN :D    :LB   :C2   :P0   7;
EQU LABELS   :D    :L1   :C2   :P8   8;
EXIT;
```

UNITS FOOT,120 means that our drawing will be in feet, with 120 system points equal to 1 foot. Since an x,y location can only be entered on a system point, the resolution to which we can specify drawing points is limited to 0.1 inches, or 1/120 foot.

LOCK 0,000 tells the system that we do not want warning messages to appear based on the angle of a line we have placed or component we have rotated.

GRID 0,250,2,0,000,0,000 indicates that we want our usable grid points (to which components can snap) to be 0.25 feet (3 inches) apart, and we want to display in the drawing area every 2nd usable grid point (meaning one point displayed every 6 inches). In addition, we want the displayed grid to lie on the xy location 0.0, 0.0.

DECIMALS 3 specifies that we want numerical outputs to show three decimal places.

FSSIZE 1,000 means our notes and text will appear 1 foot (1 user unit) high. While this might seem large, keep in mind that our floor plan will most likely be plotted at a reduced scale rather than at actual size. For example, when plotted at a scale of 0.1, a 3-by-5 foot desk is only 0.3-by-0.5 feet in size on the drawing, and the text is 0.1 feet high (slightly more than 1 inch high).

TSLANT 0,0000 means that the default slant for note and text will be zero. The slant of notes and text is measure clockwise from an imaginary vertical line.

RESOLUTION 0,0 means that arcs and circles will be drawn with as fine a resolution as possible. If we wanted to speed up drawing time, we could specify less resolution (up to 120 degrees).

DIM\_FSSIZE 1,000 means all dimension notes will appear 1 foot (1 user unit) high.

DIM\_TFSSIZE 1,000 means all dimension tolerance notes will appear 1 foot (1 user unit) high.

DIM\_TSLANT 0,0000 means all dimension notes will not be slanted.

DIM\_DEC 3 means that any dimension drawn in the normal mode will have 3 decimal places of accuracy displayed.

DIM\_SCALE 1,000 is the scale of the geometry being dimensioned. Usually this value is 1.0, but in some cases (i.e., if you are dimensioning a scaled library part) you may want to change it.

DIM\_RADIX "," means that the radix value for dimensions will be a period. In Europe this might be changed to a comma.

## 56 Customizing Your System

`DIM_EXTEND 1.000` means that the distance between the dimension points on the geometry and the extension line is 1 foot.

`DIM_DIR HORIZONTAL` means that all dimension notes will be horizontal.

`LEVEL 1` means that the default logical level is 1.

`DIM_UNITS FEET/INCHES` means that all dimension values which measure distances will be displayed in feet and inches.

`INAME :L1 :F1.000 :T1001` is only used for compatibility with EGS/45 and is not used in this example.

`TNAME :L1 :F1.000 :T1004` is only used for compatibility with EGS/45 and is not used in this example.

Layer 0 defines the instance layer. This layer is always required by the system. The layer is color #1 (white), has a line style of 8 (short dotted), and will be plotted with pen 1. Its name is "INSTBOUN".

Layer 1 defines the layer on which all desks will be placed. The layer is color #2 (red), has a line style of 1 (solid), and will be plotted with pen 1. Its name is "DESK".

Layer 2 defines the layer on which all chairs will be placed. The layer is color #3 (yellow), has a line style of 1 (solid), and will be plotted with pen 2. Its name is "CHAIR".

Layer 3 defines the layer on which all tables will be placed. The layer is color #4 (green), has a line style of 1 (solid), and will be plotted with pen 3. Its name is "TABLE".

Layer 4 defines the layer on which all cabinets will be placed. The layer is color #5 (cyan), has a line style of 1 (solid), and will be plotted with pen 4. Its name is "CABINET".

Layer 5 defines the layer on which all walls will be placed. The layer is color #6 (blue), has a line style of 1 (solid), and will be plotted with pen 5. Its name is "WALLS".

Layer 6 defines the layer on which all phones will be placed. The layer is color #7 (magenta), has a line style of 1 (solid), and will be plotted with pen 6. Its name is "PHONES".

Layer 7 defines the layer on which all extensions for open furniture will be placed. The layer is color #2 (red), has a line style of 8 (short dotted), and will be plotted with pen 0. Its name is "OPENFURN".

LAYER 8 defines the layer on which all labels notes and text will be placed. The layer is color #2 (red), has a line style of 1 (solid), and will be plotted with pen 8. Its name is "LABELS".

All of the layers are defined as DETAILED.

Had we planned to use HP-supplied library parts, we could have avoided using layers 1 through 100 which HP plans to use in current and future versions of HP EGS.

## Building or Modifying a Screen Menu

The Graphics Editor permits you to use two types of menus: screen menus and tablet menus. The use of either type of menu makes the job of entering information much easier because they replace the keyboard for most commands and functions. You may have either or both types of menus active at any time. In this section, we will discuss building and modifying a screen menu. Tablet menus will be covered in a later section.

A screen menu is convenient to use in several ways. First, it can automatically supply all the information associated with any chosen menu item, thus reducing the number of commands, command options, and part names you must remember. Second, you need not turn away from the screen when selecting an item. And third, because you have a screen menu you may also have a second viewport to show an overview or close-up of a drawing. The only real drawback of a screen menu is that it will reduce the size of the drawing area and it may only contain textual information.

A screen menu is defined by a text file contained within your personality volume or directory with the suffix `_MENU.TEXT`. The Pascal Editor must be used to create the initial template of a screen menu. This template describes the size of each box and the number of boxes contained in the menu. Once you have finished creating the template, you must save it.

There is another way to partially modify a screen menu. To add or change items in an existing menu, you can use the MENU command of the Graphics Editor (type MENU and use the stylus to pick the menu box you wish to replace). You may also place the screen menu in different corners of your display screen (type MENU 1; MENU 2; MENU 3; or MENU 4;). Note that you must save your personality's screen menu file before leaving the Graphics Editor to make your changes permanent.

---

### Note

If you use the MENU command to add words to a screen menu, the system truncates any word that is too long to fit in a menu slot. Since the information in a menu slot is transferred exactly as displayed to the Graphics Editor, you must make sure that any truncated word contained in the screen menu matches the command, macro, or name it was intended to be.

---

## Screen Menu File Structure

In a screen menu file, you may specify the location of the menu on the screen, the direction of the menu (horizontal or vertical), the color and exact location of each menu entry, and if you want the command you've picked to be displayed.

The first line of a screen menu file defines the location of the menu on the screen. The words `upper left`, `upper right`, `lower left` or `lower right` denote the corner of the screen where you want to place the menu.

Subsequent lines of the file contain the rows of the menu itself. The menu's layout in the file closely matches its appearance on the screen. A vertical bar `|` separates adjacent entries in a row. The location of these bars in the menu file determines where vertical lines will be drawn to separate items on the screen.

---

### Note

To type a vertical bar, press **Shift**  on the numeric pad of large keyboards (i.e., the HP 98203A keyboard), and **Shift**  on small keyboards (i.e., the HP 46020A keyboard).

---

The first characters in each line determine the color of each entry in that row on the screen. Colors are defined with the letters `b`, `c`, `g`, `m`, `r`, `y`, and `w`, which represent blue, cyan, green, magenta, red, yellow, and white, respectively. Each of these letters supplies a color for one menu item. However, if the number of colors specified is less than the number of slots in the line, the last color specified is assigned to all remaining slots.

Each menu row is represented by a single line in the menu file. Each row of the menu may have a variable number of slots, with individual slots containing up to 80 characters. Rows must be the same length when displayed. Thus, if you have variable length lines in your file, the system automatically adjusts the length of each line to that of the longest line by adding blank spaces to the short lines.

You can display the current primary command you are using by placing the reserved word `CURCMD` in any menu slot.

The resulting menu may be either horizontal or vertical. What determines if a menu is horizontal or vertical is the height and width of the menu.

If the height of the menu is larger than the width, the menu will be vertical. If there is sufficient space remaining on the screen, a second viewport will automatically be created. If the vertical menu is on the lower left or lower right side of the screen, the resulting second viewport will be on the top of the menu and screen. Otherwise the resulting second viewport will be on the bottom of the menu and screen.

If the width of the menu is larger than the height, the menu will be horizontal. If there is sufficient space remaining on the screen, a second viewport will automatically be created. If the horizontal menu is in the lower left or upper left side of the screen, the resulting second viewport will be on the right hand side of the menu and screen. Otherwise the resulting second viewport will be on the left hand side of the menu and screen.

## How The System Interprets Screen Menu Items

When you select a menu item from a screen menu, the system will interpret the menu item information on one of two ways:

- If the menu item is a system command or macro, the Graphics Editor understands and performs the command or macro. Commands and macros are case insensitive.
- If the menu item contains other information, the system will interpret the input literally. The difference between keyboard input and screen menu input is that `ENTER` does not appear at the end of each item selection. This is important for menus which contain number pads. If the system were to put an `ENTER` at the end of each selection, a number such as 25 could not be entered through the screen menu. Likewise, if a library part name fills an entire slot (no spaces at the end), the system will not load that part until a space, comma, or `ENTER` has been sent to it.

## Space Planning Example

The screen menu file for the example Space Planning personality is:

```
ENSP:SP_MN.TEXT
```

This file contains the following information:

```
lower right
wwwww  ! $GR! $VR! $IN! $P! $R!
w      | Edit_Drawing |
gw     | ADD*   | CURCMD |
gg     | MOVE*  | DELETE*|
gg     | COPY*  | WRAP*  |
gg     | UNDO*  | SMASH* |
gg     | ROTATE*| MODNOTE*|
gg     |         |         |
gg     |         |         |
gg     |         |         |
gg     |         |         |
gg     |         |         |
gg     |         |         |
gg     |         |         |
gg     |         |         |
gg     |         |         |
wcccccc| ENT! 0! 1! 2! 3! 4! !+!
wcccccc| EOC! 5! 6! 7! 8! 9! !-!
gg     |         |         |
yy     | OPEN*  | CLOSE* |
yy     | PAN_LFT| PAN_RT |
yy     | PAN_UP | PAN_DWN|
yy     | ZOOM_IN| ZOOM_OUT|
yy     | GRID_OFF| WINDOW*|
ggg    | IDE*  | AREA*  | DUMP*  |
ggg    | DIST* | PLOT*  | GRID*  |
yy     |         |         |
yy     |         |         |
```

This example specifies that the menu will appear in the lower right corner of the display screen, with the left over area in the top right corner dedicated to a second viewport. This menu location is usually convenient for a right-handed person working with a graphics tablet. Since the menu is longer than it is wide, it will be vertical.



All drawing commands will appear in green. Command options will appear in green beneath the commands, unless another color is specified in the macro file. These options will appear on the screen only after a particular command has been selected. Numerical and symbolic entries will appear in cyan.

The frequently used EOC (end-of-command) and ENT slots will be white for good visibility. ENT, which stands for ENTER, is a special key word for screen menus. It responds the same as pressing the ENTER button on the keyboard.

Snapping options and the current command will appear in white. These color choices match those used in other HP EGS personalities.

For convenience snapping options and drawing commands will appear at the top of the menu, and information/drawing environment/output commands will appear near the bottom. Command options (which we must specify in the macros for the different menu items) will be added when appropriate immediately below the drawing commands. Numbers, symbols, ENT and EOC will appear near the bottom of the menu.

Macros designed to manipulate the drawing on the display screen will appear in yellow near the bottom of the menu, and macros to obtain and output drawing information will appear below them in green.

All of the command macros have \* at the end of them. This indicates that they are macros, not system commands. Macros are used here instead of system commands because our personality will prompt and lead the user to do what is necessary to make the command work. HP EGS commands by themselves do not prompt and will not help the user like these macros can.

There is plenty of space left in this menu to add other commands, options, and part names at a later date. In fact, if our library of parts grows substantially in the future, we might want to design a second screen menu showing part names. We could then add a macro that would call up the second menu when we want it.

## Building or Modifying a Macro File

HP EGS allows you to create new commands, called macros, for the Graphics Editor. A command macro simply combines commands and other macros, much like shorthand notation provides a simple way of representing words and phrases. Macros can speed command entry for commands that are often entered together. And they can isolate you or other users from the actual syntax of the Graphics Editor, reducing the number of commands you need to know. In fact most users create macros solely to make commands and options more intuitive, simply renaming commands and options or adding prompts (screen messages) to help the user use a command.

A macro file is a text file stored with the suffix `_mc.TEXT`. It contains one or more `DEFINE` commands and ends with the `EXIT` command. The system allows only one macro file to exist in memory at a time. Loading another macro file will cause all existing macros to be erased.

You may use either the Pascal Editor or the Graphics Editor to create new command macros. Before you decide which method to use, consider that when you save a macro file with the Graphics Editor, the system automatically does some syntax checking and alphabetically orders your file. If you plan to write many macros or if you want to control the order of items listed in your macro file, it is best to use the Pascal Editor. However, if you intend to use a macro just temporarily while you are working on a drawing, you should use the `DEFINE` command while in the Graphics Editor. Remember, any macros which are created using the Graphics Editor are only saved in memory. If you want to save the macro permanently, use the `SAVE MACRO` command.

If you use the Pascal Editor to create your file, you must be very careful. You may load your macro file into the Graphics Editor using the `LOAD MACRO` command, but if the syntax of the `DEFINE` commands in the file is not correct, the macro file will not load into the Graphics Editor. All syntax errors must be corrected before a macro file can be loaded.

### Defining a Macro

The `DEFINE` command lets you create macros composed of Graphics Editor commands, other previously defined macros, and mathematical expressions. A macro definition involves specifying:

- the name of the macro,
- prompts or messages that you want to appear on the display screen,
- parameter names (i.e., dummy variables) for any parameters or values required by the macro,
- optional default values for these parameters, and
- the actual strings of commands and parameters which define the body of the macro.

Within a macro definition the following syntax rules apply:

- The name of the macro cannot duplicate the name of a command, library part, or layer label. If the macro name entered is the same as an existing macro, the system will replace the old definition with the new one. Macro names are case insensitive, meaning that the system would interpret `macro1` and `MACRO1` as the same thing.
- Screen prompts must be immediately surrounded by pairs of single, double or back quotes.
- Screen prompts and parameters required by the macro must appear between brackets. Within these brackets, each required parameter must be separated by a semicolon. Note that parameters *are* case sensitive.
- The body of the macro (containing strings of commands) must appear between pairs of single, double, or back quotes. Parameters required by the body of the macro must be surrounded by individual pairs of brackets without any extra spacing.
- The macro body may be made up of as many strings (250 characters maximum for each string) as is needed to define the macro.
- The entire macro definition must end with a semicolon.

For more information on using the `DEFINE` command, please refer to the *HP EGS Syntax Reference*.

## Short Macros vs. Long Macros

There are two basic types of macros: short and long. A short macro is any macro which has a body limited to one string of 250 characters maximum. These macros have a special property, as they may be used as constants in expressions. A short macro is also one in which the body of the macro can be numeric. This allows you to create constants and variables using macros. Note that the names of short macros used in expressions cannot have special characters like `*`, `-`, or `+` since these symbols are considered numeric operators.

A long macro is simply a macro that has more than one string. It uses more memory than a short macro, but has the advantage of being as long as you wish. It can be like writing a small program. Long macros are not allowed in expressions.

An example of a short macro would be

```
DEF ABC "250 CHARACTERS MAXIMUM " ;
```

while a long macro would look like this:

```
DEF XYZ "10 CHARACTERS "  
      "20 CHARACTERS " ;
```

Note that if you type a macro body from the keyboard (rather than inputting it from a file), entering a `<cr>` will terminate a string just like a closing quote, back quote, or double quote would.

## How the System Interprets Information

To understand how HP EGS interprets the contents of a macro file, you must also understand how the system reads data. The HP EGS system includes a section of code which filters and reads all data sent to it by the user. This code performs the initial processing of data as it is entered into the system either by typing from the keyboard or by selecting menu items.

The first thing this filter does is break the input into small pieces of data called tokens. A token is merely a unit of input. You can think of the words of a sentence as its tokens. In HP EGS, the tokens are the individual pieces of information entered for every command.

The second thing the filter does is classify each token. For example the token ADD is classified as a command. At any time, the system accepts only a limited number of types of input. In fact when you fail to enter an appropriate item, the filter responds with a message telling you what it expected.

When the filter encounters a macro (a name that has been defined), the following steps occur:

1. The filter looks for the macro definition in the macro file stored in current memory.
2. The filter determines how many parameters are associated with this macro.
3. The filter displays any prompts associated with the parameters.
4. The filter collects the same number of tokens as there are parameters. Note that the filter does not check to see if the tokens are appropriate values for the parameters. A space, comma, or `<cr>` is a delimiter for these tokens. A semicolon as well as a Graphics Editor command will terminate the token string. If there are not enough tokens for all the parameters defined before the end of the token string, a default value is substituted for those parameters without corresponding values.

---

### Note

A parameter cannot contain delimiters because the system uses delimiters to break the input into multiple tokens.

---

5. The filter substitutes the tokens it just collected for the parameters listed in the body of the macro, and it re-scans the resulting string of commands.

A final note to remember. When the system encounters an error, it will display a message on the screen telling you what went wrong. In addition, it will dump the most recent input lines (up to 80 characters) on the alpha screen and place a ^ symbol to indicate where the error occurred.

## Macro Substitutes for Commands

One way to make commands easier to use and more intuitively obvious to the user is to substitute macros for commands themselves. Consider the following examples:

```
DEF GRID*
  {"Enter the dist between grid points, how often to display them"} " GRID ";
```

This example defines a new command macro named GRID\*. This macro does nothing more than display a prompt on the display screen to remind you how to use the GRID command, and then it executes a normal GRID command. The \* at the end of the macro name ensures that the system will be able to distinguish between the Graphics Editor command GRID and the macro GRID\*.

```
DEF $FIT " :F ";
```

This example replaces the command option :F, which could be used in the WINDOW command, with words that more clearly describe the action you want the system to perform.

```
DEF log_on {"Enter your name" N}
  "PRE #5:/USERS/{N}";
```

This example provides a simple way for users who share an SRM to log on every day. Rather than having to type in the full path name to prefix to one's own directory, the user needs only to type in his or her name. This macro is particularly convenient for users who may be unfamiliar with SRM notation.

Of course, if you plan to use these macros often, you could place them in the screen menu using the MENU command.

## Temporary Macros

Now let's consider a macro that might be defined for temporary use during an editing session. A common situation that arises is the need to move a variety of components from one layer to another. The `MODIFY` command is used for this purpose. However, the syntax of the command makes it difficult to type in the command over and over again. Thus, we might want to create a new macro to help us perform this step repeatedly. Once the operation is complete, we probably will not need this macro again since it was tailored to the unique problem being solved at the time.

The following example shows how we could move components from the layer they are on to layer 10:

```
DEF change_it {"Select component to change"X;Y} "MOD {X};{Y} 0;0 ;L10;"
```

This macro moves a selected component from its current layer to layer 10, with no change in position. To call up this macro, we could type in its name (or place the name in the menu and select it), and then select the component. The `MODIFY` command requires that we enter two x,y points. The first selects the item to be modified; the second defines a reference point which is used if the item is moved. Since we only want to change the layer number, a dummy reference point 0,0 appears in the body of the macro; thus we only have to pick one point when we invoke the macro.

Let's see what sequence of events really happens when we call up the macro `change_it`:

1. The prompt `Select component to change` appears at the top of the display screen.
2. The filter waits for us to give it the next two tokens. These should be the coordinates of the object that we select. Let's assume that we select a point whose coordinates are 100,200.
3. The filter substitutes these values for the parameters in the body of the macro and re-scans the replacement string. The final result is the same as if we had typed in

```
MOD 100,200 0;0 ;L10;
```

To `MODIFY` another component, we must invoke the macro again. If we have a large keyboard (i.e., HP 98203A), we can do this conveniently by pressing **f2** (the **RECALL** softkey) (assuming that the last thing we entered was the macro name and it was typed from the keyboard) and **Return**. Then we can select the next item to move. Alternatively, we could place this macro in the screen menu and repeatedly select it.

Suppose now that we want to move *all* rectangles on layer 5 to layer 7. Although we could do this like we did the previous macro, it is easier to define a macro that does all the moves for us at once. Consider the following macro:

```
DEF change_all "MOD R5 0;0 0;0 ;L7; change_all ";
```

This macro moves all rectangles on layer 5 to layer 7 by continually calling up the same macro until all rectangles have been moved. Thus by invoking a simple macro once, we let the system perform all of the work. Note that the last thing the macro does is to call itself. Any macro which calls itself again is termed recursive.

This recursive macro makes several assumptions. First, it assumes the point 0,0 is in the display viewport; second, it assumes all the rectangles we want to move are also in the display viewport. If they are not, we could easily move them into the display using the WINDOW :F command. Note that the descriptor R5 limits the search to rectangles on layer 5. Had this descriptor been left out, the macro would loop forever. If you accidentally do write a macro which loops forever, press **Stop** until the system gives you an error message. In general, recursive macros do not stop and are of limited use. However, if the system encounters an error, it terminates the macro. This macro causes an error, since eventually there must be no more rectangles on layer 5 to modify.

## Macros to Save Values and Set Modes

Macros can also help you save values and set modes. For example, in a floor plan we might want to label each desk with a sequential number such as DESK1, DESK2, and so on. Of course, it would be convenient if we could select the item and have it automatically labeled with the next sequential number. The following macros do this:

```
DEF VAL 1;
DEF CLASS "DESK";
DEF TIC "'";
DEF DESK_LABEL {"Select location for desk number"X;Y}
  "ADD NB (TIC+CLASS+NUM2STR(VAL)+TIC) {X},{Y}; NOP
  DEF VAL (VAL+1);";
```

Let's take a moment to look at how this routine works.

- The macro VAL defines a numeric constant for later use as a variable in an expression. VAL indicates the initial value for the desk number, which in this case is 1.
- The macro CLASS defines the alpha portion of the label, which is DESK. Since the definition is enclosed in quotes and it is a short macro, the value is a text string.
- Similarly, the macro TIC defines a constant whose value is the single quote character. This is done to prevent problems with quotes within quotes in the next macro.
- The macro DESK\_LABEL is the heart of the solution. It accepts an x,y location for the desk number and adds the note defined inside the parentheses on layer 6. The note consists of a TIC ( ' ), a CLASS (DESK), the number represented by VAL (initially 1), and a second TIC. The NUM2STR function is allowed in an evaluated expression (see the EVALUATE command in the *HP EGS Syntax Reference*). The result of this macro is the note DESK1 added to layer 6. Next the NOP command returns the system to a no-command state. To increase the next number in the sequence by one, VAL is redefined to be VAL+1. The semicolon after the entire macro terminates the command.

## Macros to Modify Existing Information

The HP EGS personalities may contain some information you want to modify. For example, the Electrical Schematic Drawing personality provides default reference designators for each library part. Suppose we want to increase the numerical values of these designators incrementally when we add them. Instead of just adding a designator, we need a macro to modify the default value. The following macros do this:

```
DEF VAL 1;
DEF CLASS "U";
DEF TIC "'";
DEF AUTOREF {"Select the reference designator to modify"X;Y}
"MOD N {X},{Y} 0,0 :E (TIC+CLASS+NUM2STR(VAL)+TIC);
DEF VAL (VAL+1);";
```

This macro operates in a manner similar to the one described previously. The major difference is that it modifies the existing designator (using MOD :E) instead of adding new text to the drawing.

## System-defined Macro Constants

To allow your macros and macro instances to interact easily with the HP EGS system, a series of constants are provided. These constants are macros whose values denote the state of the system. Some of these constants define environment values while others relate to the most recent component selected in a drawing. You can use these values in expressions as needed. However, you should not use these names for your own macro constants, as the system sets the values corresponding to these names and would overwrite any values that you might store.

### System Constants

The following constants define the current system status:

P1X	Coordinates of the lower left and upper right corners of the current window for viewport 1
P1Y	
P2X	
P2Y	
SYSTEM_AREA	The value returned by the AREA command
SYSTEM_ASSD_ON_PORT1	1 if associated text in port 1 is turned on, 0 if off
SYSTEM_ASSD_ON_PORT2	1 if associated text in port 2 is turned on, 0 if off
SYSTEM_CUM_AREA	The cumulative area returned by the AREA command
SYSTEM_DECIMALS	Number of decimal places for output
SYSTEM_DIM_DECIMALS	Number of decimal places for output of dimensions
SYSTEM_DIM_DIR	Current orientation of dimensions (HORIZONTAL or IN-LINE)

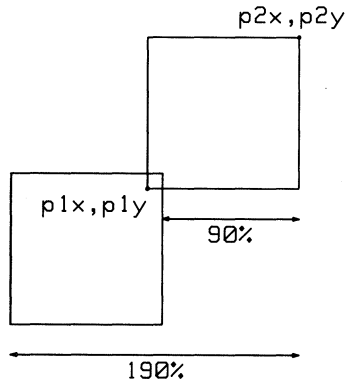


SYSTEM_DIM_EXT	Dimension extension distance
SYSTEM_DIM_FSIZE	Default character height for text in dimensions
SYSTEM_DIM_SCALE	Current scale factor applied to dimensions
SYSTEM_DIM_UNITS	Current units of the dimensions
SYSTEM_DIST	The vector distance between the two points selected for the DISTANCE command
SYSTEM_DIST_X	The X distance between the two points selected for the DISTANCE command
SYSTEM_DIST_Y	The Y distance between the two points selected for the DISTANCE command
SYSTEM_DRAWING_NAME	The name of the current drawing
SYSTEM_ECHO	The value indicating the current status of the ECHO command (1 = ECHO ON, 0 = ECHO OFF)
SYSTEM_FSIZE	Default character height for notes and text
SYSTEM_GRID_ON_PORT1	1 if grid in port 1 is turned on, 0 if off
SYSTEM_GRID_ON_PORT2	1 if grid in port 2 is turned on, 0 if off
SYSTEM_GRID_RES	Current number of grid points per user unit
SYSTEM_GRID_SPACE	How often grid points are displayed
SYSTEM_LEVEL	The current logical level
SYSTEM_LOCK	The current lock angle
SYSTEM_LOCK_ENFORCE	0 if NOT enforced, 1 if enforced
SYSTEM_NEST_PORT_1	The current nesting level in viewport 1
SYSTEM_NEST_PORT_2	The current nesting level in viewport 2
SYSTEM_NVOLS	The number of volumes in the volume search table
SYSTEM_POINT_X	The X coordinate value returned by the POINT command
SYSTEM_POINT_Y	The Y coordinate value returned by the POINT command
SYSTEM_RADIX	Radix used for numbers ( . or , )
SYSTEM_RAW_PNT_X	The raw x,y coordinates of the last point selected
SYSTEM_RAW_PNT_Y	
SYSTEM_SNAP_MODE	The current snapping mode (:GRID, :RAWPOINT, :INTERSECT, :VERTEX, :PRIMITIVE)
SYSTEM_SNAP_PNT_X	The x,y coordinates of the last point snapped to
SYSTEM_SNAP_PNT_Y	
SYSTEM_UNIT_RES	The resolution of the units
SYSTEM_UNIT_TYPE	The type of units currently in use (inches, feet, etc.)

Let's take a look at an example. HP EGS automatically defines four system constants for your use –  $p1x$ ,  $p1y$ ,  $p2x$ , and  $p2y$ . They are the coordinates of the lower left and upper right corners of viewport 1. These are used by the personalities to define the panning macros. One of these macros is as follows:

```
DEF PAN_LL "WIN (P2X-1,9*(P2X-P1X)), (P2Y-1,9*(P2Y-P1Y))
            (P2X-0,9*(P2X-P1X)), (P2Y-0,9*(P2Y-P1Y));"
```

The effect of this macro is to move the viewport to the left and down. These equations define the two diagonal corners of the new rectangle based on the location of the point  $p2x, p2y$ . The new coordinates are 190% and 90% to the left and down from  $p2x, p2y$ . Thus, the area displayed is the same size as the previous viewport and overlaps it by 10% in both the x and y directions.



### Component Constants

Each time a component is selected in the system, a series of macro constants is set. They are set whenever a component is selected in a drawing. Unless specified otherwise, the value is expressed in user units.

### All Components

The following constants are defined regardless of the type of component referenced:

SYSTEM_ASSOC_TEXT	If the component has associated text, this string contains that text
SYSTEM_LAYER	Numeric value defining the layer (1-255) of the component; zero is used for instances

SYSTEM\_LL\_X Defines the boundary rectangle for the selected  
 SYSTEM\_LL\_Y component (lower left and upper right coordinates)  
 SYSTEM\_UR\_X  
 SYSTEM\_UR\_Y

SYSTEM\_PRIMITIVE Defines the type of the last component (primitive)  
 referenced. Is one of the following: ARC, CIRCLE,  
 DIMENSION, HATCH, INSTANCE, LINE, MARKER,  
 NOTE, OVAL, POLYGON, RECTANGLE, TEXT

SYSTEM\_STEP\_COLS If the component is implicitly stepped, these  
 SYSTEM\_STEP\_ROWS constants have the step values  
 SYSTEM\_STEP\_ROTATION

### Arc

SYSTEM\_CENTER\_X The x,y location of the center of the arc  
 SYSTEM\_CENTER\_Y  
 SYSTEM\_FILL 0 if the arc is unfilled, 1 if filled  
 SYSTEM\_FINISH\_ANGLE The ending angle of the arc  
 SYSTEM\_RADIUS The radius of the arc  
 SYSTEM\_RESOLUTION The resolution of the arc  
 SYSTEM\_START\_ANGLE The initial angle of the arc  
 SYSTEM\_WIDTH The width of the arc

### Circle

SYSTEM\_CENTER\_X The x,y location of the center of the circle  
 SYSTEM\_CENTER\_Y  
 SYSTEM\_FILL 0 if the circle is unfilled, 1 if filled  
 SYSTEM\_RADIUS The radius of the circle  
 SYSTEM\_RESOLUTION The resolution of the circle  
 SYSTEM\_WIDTH The width of the circle

### Dimension

SYSTEM\_DIM\_PT1\_X The end points of a linear dimension  
 SYSTEM\_DIM\_PT1\_Y  
 SYSTEM\_DIM\_PT2\_X  
 SYSTEM\_DIM\_PT2\_Y  
 SYSTEM\_DIM\_SLANT The slant of the dimension text  
 SYSTEM\_TOL\_FSIZE The font size of the tolerance values

### Hatch

SYSTEM\_ROTATION           The rotation of the hatch lines  
 SYSTEM\_WIDTH             The spacing between hatch lines

### Instance

SYSTEM\_INST\_NAME         The name of the instance, null if unnamed  
 SYSTEM\_MIRROR           The mirror of the instance (:MX, :MY, :MXY, NOMIRROR)  
 SYSTEM\_ORIGIN\_X         The x,y location of the instance  
 SYSTEM\_ORIGIN\_Y  
 SYSTEM\_SCALE             The scale of the instance  
 SYSTEM\_ROTATION         The rotation of the instance

### Line, Polygon

SYSTEM\_CONST\_LINE        0 if line is NOT construction line, 1 if it is  
 SYSTEM\_FILL              0 in unfilled, 1 if filled  
 SYSTEM\_LEADER\_LINE      0 if line is NOT leader line, 1 if it is  
 SYSTEM\_WIDTH            The width of the line (polygon boundary)

### Marker

SYSTEM\_ORIGIN\_X         The x,y location of the marker  
 SYSTEM\_ORIGIN\_Y

### Note, Text

SYSTEM\_FILL              0 if text unfilled, 1 if filled  
 SYSTEM\_FONT\_SIZE        The height of the characters  
 SYSTEM\_FONT\_SLANT       The slant of the characters  
 SYSTEM\_LOG\_LOC\_X        The logical x,y location of associated text  
 SYSTEM\_LOG\_LOC\_Y  
 SYSTEM\_MIRROR           The mirror of the text (:MX, :MY, :MXY, NOMIRROR)  
 SYSTEM\_ORIGIN\_X         The x,y location of the text  
 SYSTEM\_ORIGIN\_Y  
 SYSTEM\_ROTATION         The rotation of the text  
 SYSTEM\_TAG               The value of the tag if this is associated text  
 SYSTEM\_TEXT             The actual text itself  
 SYSTEM\_TEXT\_REMAIN      The remainder of a text string after initial  
                           numerical values are removed  
 SYSTEM\_TEXT\_SLANT       The slant of the text  
 SYSTEM\_TEXT\_VALUE       The initial numerical values contained in a text string

## Oval

SYSTEM_CENTER_X	The x,y location of the center of the oval
SYSTEM_CENTER_Y	
SYSTEM_FILL	0 if the oval is unfilled, 1 if filled
SYSTEM_FINISH_ANGLE	The ending angle of the oval
SYSTEM_RADIUS	The radius of the oval
SYSTEM_RESOLUTION	The resolution of the oval
SYSTEM_START_ANGLE	The initial angle of the oval
SYSTEM_WIDTH	The width of the oval

## Rectangle

SYSTEM_FILL	0 if unfilled, 1 if filled
SYSTEM_WIDTH	The width of the rectangle

Let's look at an example that uses a system constant to find out the current snapping mode, and uses system constants to define the location of the boundary rectangle surrounding a component. Suppose we want to add a circle around selected components in a drawing. We want to place each circle the same distance from the edge of each component, such that all encircled components in a drawing look similar. This means we must use the rawpoint snapping mode to place the circles. We could use the macro:

```
DEF ENCIRCLE {"Select the component to encircle," X;Y}
  IDENTIFY {X},{Y};
  DEF OLDSNAPMODE (''+SYSTEM_SNAP_MODE+''); :RAW "
  ADD C1 :RC "
  "(0.6*SQRT(SQR(SYSTEM_UR_X-SYSTEM_LL_X)+SQR(SYSTEM_UR_Y-SYSTEM_LL_Y))) "
  "((SYSTEM_UR_X+SYSTEM_LL_X)/2) , ((SYSTEM_UR_Y+SYSTEM_LL_Y)/2);
  OLDSNAPMODE NOP ";
```

This macro performs the following:

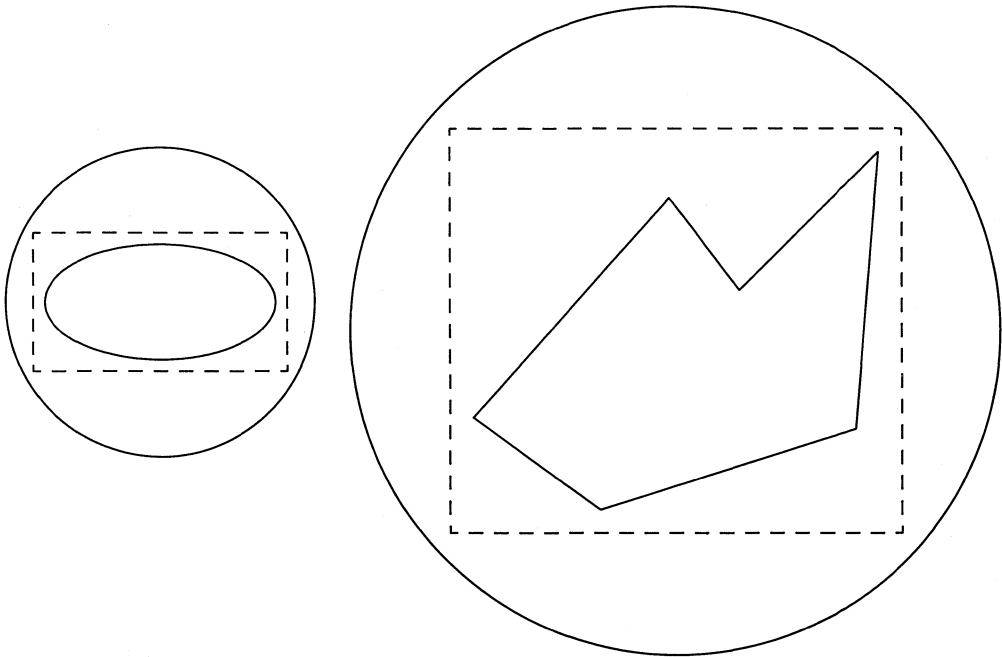
1. It defines the macro ENCIRCLE, the prompt that should appear on the screen when this macro is selected, and the x,y coordinates that identify the selected component.
2. The IDENTIFY command is used to set the system constants for the selected component. These constants include the upper right and lower left corners of the boundary rectangle.
3. A new macro OLDSNAPMODE is defined to record the current snapping mode. This allows us to reset the snapping mode to its old value after we are done using the ENCIRCLE macro. Notice how the quote marks around the snapping mode are appended to the text string (using quotes within quotes); quote marks by themselves would be ignored by the system. Note also that quote marks are necessary to distinguish text strings from integers or real numbers in evaluated expressions.

4. The `:RAW` snapping mode is set so that we can add the circle exactly where we want it (independent of the grid setting or location of other components in the drawing).
5. A circle is added to layer 1 with the radius-centerpoint method. To find the radius, we use the Pythagorean theorem to calculate the distance from the lower left corner of the boundary rectangle to the upper right corner (which forms a diagonal line through the rectangle). We multiply this value by 0.6 so that the circle, once drawn, will appear a little bigger than the boundary rectangle being encircled (multiplying by 0.5 would have placed the circle right on the corners of the boundary rectangle itself).

To find the center of the circle, we average the distances of both sides of the boundary rectangle and divide them by two. These values give us the x,y coordinates of the centerpoint of the boundary rectangle.

6. Once the circle is added, we invoke the macro `OLDSNAPMODE` to return to the snapping mode that was previously in use. We terminate the `ADD` command with `NOP` which places the graphics Editor into a no-command state.

Note that if you try to type in this macro as it appears on this page, you must pay attention to spacing. Although some quotes appearing in this macro are not required, they are added to emphasize where the extra spaces are necessary.



## Conditional Macros

In some situations, the effect of a macro should depend on the data. The macro function `IFELSE` can be used for this purpose. The `IFELSE` function considers three parameters:

- The condition to evaluate. The condition must be true or false. Whenever HP EGS returns a numeric result, 0 is considered false and all other values are considered true.
- The value to return if the condition is true.
- The value to return if the condition is false.

The following example illustrates the concept. Suppose you wish to have a single macro that turns the grid on or off. Recall that even if the grid is turned on, it may not be visible. This could be because the grid is too dense, or because it is very coarse and you are “windowed in.” Suppose we also wish to display the current grid setting (on or off). The following three macros accomplish this task. Let us assume that menu slot 3,1 is being used to display the grid setting information.

```
DEF $rOFF {"The grid will be turned on"} " switch '$rON' 0 ";

DEF $rON  {"The grid will be turned off"} " switch '$rOFF' 1 ";

DEF switch {M;C}
  "MEN :C=Y {M} 3,1; (IFELSE (SYSTEM_GRID_ON_PORT1={C}, ' WIN :G;', ' ')) ";
```

In the first two macros, we define `$rON` and `$rOFF` such that when one of them is selected, a prompt appears on the screen and the action `switch` takes place. The action `switch` is defined in the third macro using two values: `M` for menu, and `C` for constant (the last two items of the previous macros). This macro then uses the `MENU` command to specify the color (yellow inverse video) and location of the text string to be placed in the screen menu. Next a condition is specified: (1) if the system constant and `C` agree (meaning the condition is true), then the system performs the action listed inside the first pair of single quotes; (2) if the system constant and `C` do not agree (meaning the condition is false), then the system performs the action listed inside the second pair of single quotes.

Let's consider how these macros might work in a real situation. Assume that initially the grid is off and the menu item `$rOFF` is displayed in the menu slot 3,1. When the item `$rOFF` is selected, the following steps take place:

1. The prompt `The grid will be turned on` is displayed.
2. The text `" switch '$rON' 0 "` is substituted for `'$rOFF'` and the process continues.
3. Since `switch` is a macro with two parameters, the filter collects two tokens (in this case the string `'$rON'` and the value `0` which are supplied by the original macro).

4. These two parameters are then substituted as defined in the `switch` macro and the filter continues.
5. As a result of these two substitutions, a `MENU` command is executed which places a macro defining the current state (namely `grON`) in the appropriate menu box.
6. Next, the filter evaluates the expression contained in parentheses. This expression uses the `IFELSE` function and compares the current grid setting (defined by the system macro constant `SYSTEM_GRID_ON`) with the second parameter of the macro `switch`. If they agree (i.e., the grid setting matches the value specified in the macro defining it), the action `WIN :G;` takes place, thus executing a `WINDOW` command and toggling the grid setting. If they do not agree, a blank string is evaluated and the command is completed.

To turn the grid off, you could select the menu box labeled `grON` and a similar process would occur, replacing the menu box with the words `grOFF`.

## Space Planning Example

The macro file for the Space Planning personality is:

```
EWSP:SP_MC,TEXT
```

This file contains the information needed for all the macros in the screen menu plus any other macros we might reference. Most of the examples in the Space Planning personality are simple and very straight forward. Below are some examples which will help in understanding this macro file.

We must add the new `OPEN*` and `CLOSE*` macros for our personality, which will display furniture with doors/drawers opened and closed, in the macro file. Remember that in the process file we defined earlier, we stated that anything added to layer 7 would represent open furniture. These macros could be written as follows:

```
DEF OPEN* {"Furniture will be opened,"}
    " SHOW #E7 ;";
```

This macro requests that all components on layer 7 be displayed.

```
DEF CLOSE* {"Furniture will be closed,"}
    "SHOW -E7;";
```

This macro requests that all components on layer 7 be turned off.

For more information on the other macros defined for this personality, see the macro file `EWSP:sp_mc.TEXT`. This file is annotated to explain what many of the macros do.



## Creating Library Parts

In many cases you may want to use the same parts over and over again in your drawings. You can create your own library of parts by saving drawings in drawing files (files ending with `.d`). Each time you bring a saved drawing into another drawing, the system will display it as an instance.

To create a part, use the `EDIT` command to call up a clear display screen. Then place the components you want on the screen and use the `SAVE` command to store the drawing.

The first thing you should check when you begin creating a part is your personality's process file. This file contains many instructions that help the system set up the drawing environment. For example, you may want to ensure that the drawing units (feet, mm, etc.) are appropriate for the part being drawn, and that there are enough system points per unit for good resolution of the drawing on the screen. In addition you may want to set the drawing grid (to which points can snap) to include just the number of system points you need, and display only those grid points that are convenient to work with.

To modify an existing part, use the `EDIT` command to bring the part up on the screen, change the part as necessary, and save the changed drawing under its original file name. Then when you call up any other drawings that reference this drawing, they will reflect the changes you made.

### Special Drawing Considerations

It is wise to consider your eventual uses of parts before you design them. This preplanning can save you much revision work later on. Think about these topics whenever you are creating new drawings and library parts:

- **Grid Setting:** Where would you like points to snap on your drawing screen? How many grid points would you like to see in the display? When you first begin drawing a part, you may want to window in to a smaller portion of the screen so that you can see your grid display and so that you draw your part at a realistic scale.
- **Drawing Scale:** If possible make your drawings to scale (i.e., actual size based on the units specified in your process file).
- **Origin Placement:** The origin (0,0 point) always becomes the reference point for a drawing. Whenever you add a part to another drawing, the system tracks the part as you move it around by this reference point. This means that the origin should be located where it is convenient for part placement in a drawing (i.e., on a point you want to line up with the grid, or in the same place as a physical or logical port).
- **Layers:** You can place various parts of a drawing on different layers if you need to display them separately as well as together. You may also bind different layers together in the process file – this lets you turn a layer on or off in conjunction with another layer.

- **Nesting Levels:** For some drawings you may want to nest parts inside parts (creating instances within instances), which allows you to display selected levels of detail on your screen at any time. This can be particularly useful for drawings where you want to show physical hierarchy. To change an instance within a drawing, you may use the EDIT command; this is convenient since you do not have to call up the drawing file itself to make the desired changes.
- **Associated Text:** When you associate text with a library part, you may associate it either to a component in the drawing or to the drawing itself. You may want to associate the text to the endpoint of a component (using vertex mode) if this is also the location of a logical point. If you associate text to a drawing, and then add that drawing to another drawing, the text is copied and associated to the instance component. You always have the option to display associated text or not; the default is to not display it since it often clutters a drawing. Tag values may be assigned to associated text if you plan to use post-processors. Tag values are numerical classifications of parts which you may define.
- **Logical Levels:** When you create a part the system automatically assigns it the logical level 1. However, you may want to assign lower levels to parts in order preserve a logical or conceptual hierarchy.
- **Post-processors:** Pay close attention to the associated text in your drawings if you plan to extract additional information about your drawings from other application programs. For example, the HP EGS connection list utilities require that annotation among schematics being merged must be unique for each part, and that annotation between a schematic and a PC board being compared must match.
- **Instances:** If you plan to create parts by nesting instances within a drawing, be careful about smashing these instances. Smashing them will destroy all drawing hierarchy. If you simply want to see more detail inside the instances you are working with, use the :N option of the WINDOW command.

## Macro Instance Library Parts

A “macro instance” is a macro that describes an instance (or library part). This instance differs from an ordinary instance because, each time you add the macro instance to a drawing, you must supply values for the macro’s parameters which determine how the macro instance is to appear.

In essence, a macro instance defines a “family of parts” whose members look alike but are not exactly the same. One common use of macro instances might be in mechanical drawing, where you could have parts classified as bolts, nuts, hinges, etc. Each time you add one of these parts to a drawing, it would be convenient to add a standard part and simply specify the values that characterize that particular part. This means you could make one library part for each type of object, instead of one for every possible size and shape of that object.

Use the Pascal Editor to define a macro instance (using the `DEFINE :I` command), and save each instance in a separate file ending with the suffix `.i`. When a drawing containing a macro instance is saved, the drawing file contains only the reference to the macro instance's file and a list of the parameters needed to create the drawing. Also make sure the name of the macro you define is exactly the same as the file name you save it under.

Once a macro instance has been added to a drawing (using the `ADD <instance>` command) and all the parameter values supplied, a graphical representation of the result is displayed. You may then manipulate the macro instance like any other instance.

The `EDIT` command allows you to change a macro instance in two ways: by changing the definition of the macro instance or by changing the values supplied for any parameters in the macro instance. For more information on editing instances, please refer to the `EDIT` command in the *HP EGS Syntax Reference*.

One example of a macro instance is for space planning. This particular one would allow you to create a desk of any size, rotated at any angle. This macro instance file contains the following information:

```
DEF :I DESK {"Enter the desk length" length "1" ;
            "Enter the desk width" width "1" ;
            "Enter the desk rotation" rotation "0"}
`DEF OLDSNAPMODE ("'+SYSTEM_SNAP_MODE+'") ;RAW `
"ADD P1 0,0 {length},0 {length},{width} 0,{width} 0,0;"
"MOD P1 0,0 0,0 :R{rotation};"
" OLDSNAPMODE " ;
```

This macro builds a macro instance of a desk as follows:

1. It defines the macro `DESK`, the prompts that should appear on the screen when this macro is selected, and the parameters (`length`, `width`, `rotation`) we need to specify to make the drawing.
2. A new macro `OLDSNAPMODE` is defined to record the current snapping mode. This allows us to reset the snapping mode to its old value after we are done using the `DESK` macro instance.
3. The `:RAW` snapping mode is set so that we can add components exactly where we want them (independent of the grid setting or location of other components in the drawing).
4. We use an `ADD` command to draw a polygon which is the length and width of the desk we wish to create. This `ADD` command refers to the parameter values `length` and `width`.
5. The `MODIFY` command is called to rotate the polygon the correct angle required.
6. Once the desk is drawn, we invoke the macro `OLDSNAPMODE` to return to the snapping mode that was previously in use. We terminate the `DEFINE` command with a semicolon.

To place this macro instance in a drawing, we could type in `ADD desk`, enter the parameters, and enter the desired x,y coordinates or select a point with the stylus. Then we could redraw the instance at a deeper nesting level to see the desk using `WIN :N2 ;` .

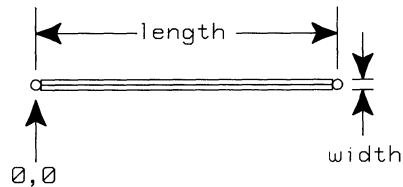
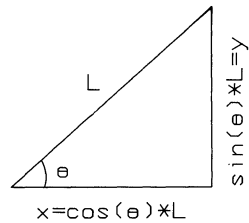
Now let's take a look at a more complex macro instance designed to draw wall dividers. There is usually a need to add partitions to the layout of any area. These partitions, or walls, may be purchased from a manufacturer and ordered in different lengths and widths. The following macro instance allows you to specify different partition lengths, widths, and angles of rotation with just one definition.

```
DEF :I WALL {"Enter the wall length" length "1" ;
            "Enter the wall width" width "1" ;
            "Enter the wall rotation" rotation "0"}
`DEF OLDSNAPMODE ("'+SYSTEM_SNAP_MODE+'"); :RAW `
"ADD C5 :RC ({width}/2)
           0,0
           ({length}*COS({rotation})), ({length}*SIN({rotation}));"
"ADD L5 :FILL :W {width}
           ({width}/2*COS({rotation})),({width}/2*SIN({rotation}))
           (({length}-{width}/2)*COS({rotation})),
           (({length}-{width}/2)*SIN({rotation}))"
" OLDSNAPMODE ";
```

This macro builds a macro instance of a wall divider as follows:

1. It defines the macro `WALL`, the prompts that should appear on the screen when this macro is selected, and the parameters we need to specify (`length width rotation`) to make the drawing.
2. A new macro `OLDSNAPMODE` is defined to record the current snapping mode. This allows us to reset the snapping mode to its old value after we are done using the `WALL` macro instance.
3. The `:RAW` snapping mode is set so that we can add components exactly where we want them (independent of the grid setting or location of other components in the drawing).
4. We use the `ADD` command to draw two circles which are at the ends of the wall we wish to create. This `ADD` command refers to the parameter values `length, width, rotation`.
5. We use a second `ADD` command to draw the wall itself. The wall will be a line with width and will be filled. This `ADD` command refers to the parameter values `length, width, rotation`.

6. Once the wall is drawn, we invoke the macro `OLDSNAPMODE` to return to the snapping mode previously in use. We terminate the `DEFINE` command with a semicolon.



Another more complex example of a macro instance appears in the HP EGS Mechanical Drafting personality, in the drawing named `EWME:nutsd.i`. This macro instance builds a nut shown from the side view as follows:

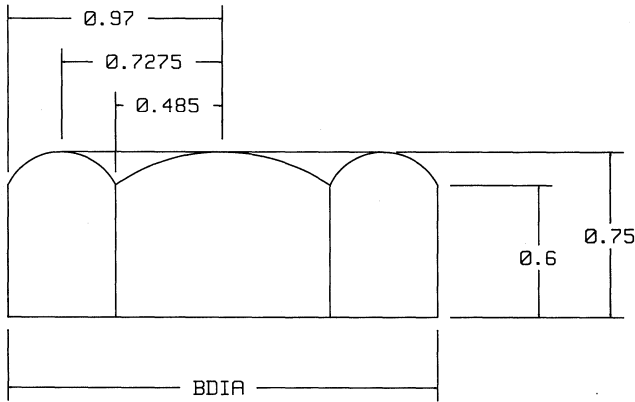
```

DEF :I NUTSD {"ENTER the nut diameter," BDIA}
`DEF OLDSNAPMODE ("'+SYSTEM_SNAP_MODE+'"); ;RAW `
"ADD A11 (0,6*{BDIA}),(0,97*{BDIA}) (0,6*{BDIA}),(0,485*{BDIA})"
" (0,75*{BDIA}),(0,7275*{BDIA}) ;"
" (0,6*{BDIA}),(0,485*{BDIA}) (0,6*{BDIA}),(-0,485*{BDIA})"
" (0,75*{BDIA}),0 ;"
" (0,6*{BDIA}),(-0,485*{BDIA}) (0,6*{BDIA}),(-0,97*{BDIA})"
" (0,75*{BDIA}),(-0,7275*{BDIA}) ;"
" L11 (0,6*{BDIA}),(-0,97*{BDIA}) 0,(-0,97*{BDIA})"
" 0,(0,97*{BDIA}) (0,6*{BDIA}),(0,97*{BDIA}) ;"
" (0,75*{BDIA}),(-0,7275*{BDIA}) (0,75*{BDIA}),(0,7275*{BDIA}) ;"
" 0,(-0,485*{BDIA}) (0,6*{BDIA}),(-0,485*{BDIA}) ;"
" 0,(0,485*{BDIA}) (0,6*{BDIA}),(0,485*{BDIA}) ;"
" OLDSNAPMODE " ;

```

1. It defines the macro `NUTSD`, the prompt that should appear on the screen when this macro is selected, and the parameter we need to specify (`BDIA`) to make the drawing.
2. A new macro `OLDSNAPMODE` is defined to record the current snapping mode. This allows us to reset the snapping mode to its old value after we are done using the `NUTSD` macro.
3. The `:RAW` snapping mode is set so that we can add components exactly where we want them (independent of the grid setting or location of other components in the drawing).
4. We use a series of `ADD` commands to draw three arcs and several lines outlining the nut (side view perspective). These `ADD` commands refer to the parameter value `BDIA`.
5. Once the nut is drawn, we invoke the macro `OLDSNAPMODE` to return to the snapping mode that was previously in use. We terminate the `DEFINE` command with a semicolon.

To place this macro instance in a drawing, we could type in `ADD nutsd_i`, specify the diameter of the nut, and enter the desired x,y coordinates or select a point with the stylus. Then we could use the `WINDOW` command to redraw the instance at a deeper nesting level to see the nut.

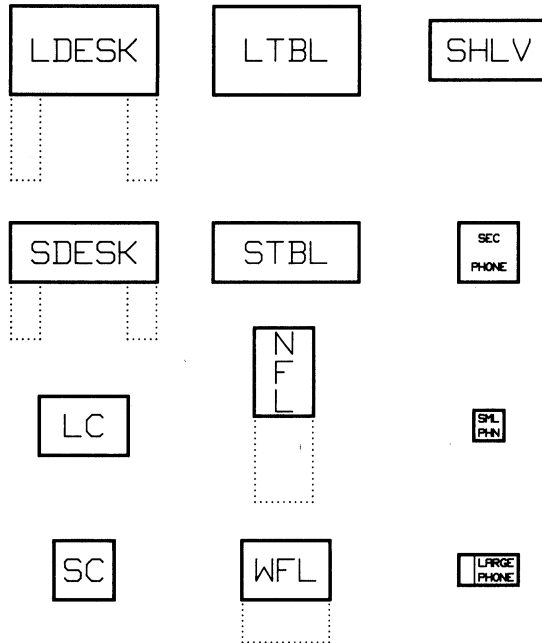


## Space Planning Example

The library parts created for the Space Planning personality are located in the volume EWSP: shipped with your system, under the following drawing file names:

EWSP:ldesk_d	large desk
EWSP:sdesk_d	small desk
EWSP:lchair_d	large chair
EWSP:schair_d	small chair
EWSP:ltable_d	large table
EWSP:stable_d	small table
EWSP:wfile_d	wide file cabinet
EWSP:nfile_d	narrow file cabinet
EWSP:shelves_d	shelves
EWSP:sphone_d	small phone
EWSP:lphone_d	large phone
EWSP:secphone_d	secretary's phone

These parts appear like this:



In creating these parts, special care was taken to place the various pieces of furniture on the different layers defined in the process file. In particular, the lines for opened desk drawers and file cabinet doors are on layer 7, which allows us to display them or not with the OPEN\* and CLOSE\* macros.

The Space Planning personality also has two macros instance library parts. They are:

```
EWSP:desk_i  
EWSP:wall_i
```

We showed you the text files defining the macro instances `desk_i` and `wall_i` earlier. Once drawn on the screen the walls would look as follows, depending on the exact parameter values you specify:

  1 FT

  2 FT

  3 FT

  4 FT

  5 FT

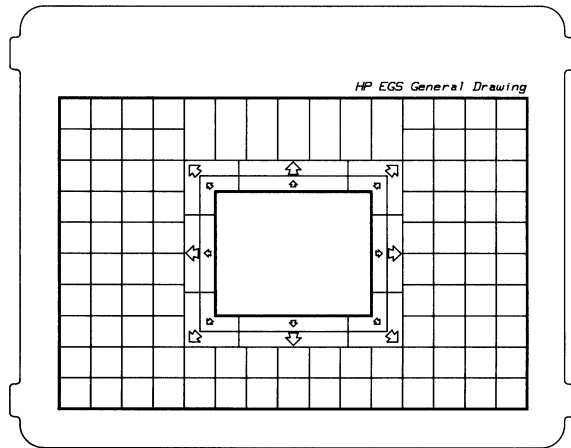
  6 FT



## Building or Modifying a Tablet Menu

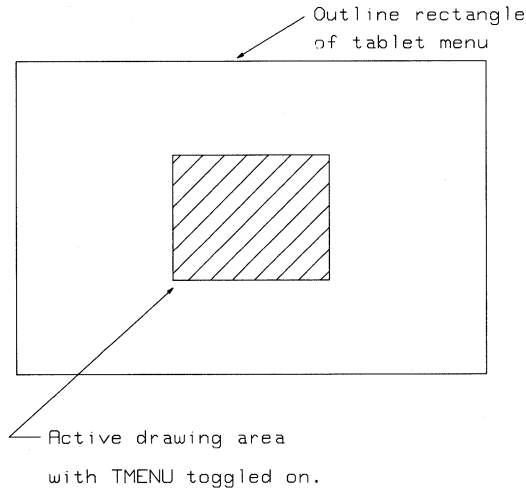
A tablet menu is a drawing that you create with the Graphics Editor, plot on paper or acetate, and overlay on the graphics tablet. It provides an alternative way to enter commands other than the screen menu or keyboard. A tablet menu not only supplements a screen menu, but can also replace it if you want to remove the screen menu and enlarge the drawing area. The tablet menu can contain both textual and graphical items.

An example of a tablet menu is shown here:

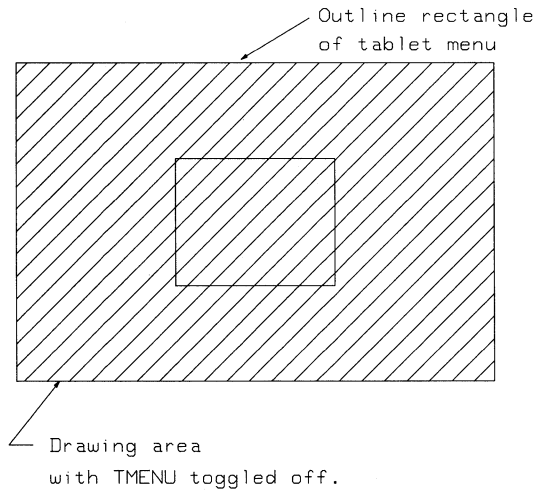


This is the HP EGS General Drawing tablet menu (contained in EWSYS:gd\_tm.TEXT and EWSYS:gdtmenu\_d). You may use this menu as a template and customize it for your application. The arrows displayed around the inner rectangle are the only menu items. When you select one of the arrows with the stylus, the system pans the screen area in the direction of the arrow. This panning is caused by macros associated with the tablet menu items.

This tablet menu can be toggled on and off by entering `TMENU :`. When the tablet menu is on, the cursor appears on the screen only when you position the stylus over the inner rectangle of the tablet menu. The area within this rectangle is called the active drawing area.



If the tablet menu is toggled off, the cursor appears on the screen when you move the stylus anywhere on the graphics tablet. The system disregards any information appearing on the plotted tablet menu.



## Drawing a Tablet Menu

Although the General Drawing tablet menu is mostly blank, other menus created for the HP EGS Mechanical Drafting, Electrical Schematic Drawing, and Printed Circuit Board Layout personalities contain library parts. For instructions on how to plot and use these menus, see Appendix A of *Learning HP EGS*.

To draw a new tablet menu for your application, you can create one from scratch or use the General Drawing tablet menu as a template. The General Drawing tablet menu is convenient to modify and is usually suitable for most applications. For instructions on modifying this tablet menu, see Appendix C of *Learning HP EGS*.

To draw a tablet menu from scratch:

1. Consider its scale. It is most convenient to draw the menu at a scale of 1 so you can avoid having to stop and figure the correct scale each time you plot the menu. The following tablet menu dimensions fit on the HP 9111A graphics tablet:

Outline Rectangle	8 by 12 inches (203 by 305 mm)
Active Drawing Area Rectangle	can be any size inside the outline

2. Draw the outline rectangle, which serves as the outside boundary of the tablet menu.
3. Enter:

```
ADD N :T1010 :AC Return
```

and select the outline rectangle.

4. Enter:

```
"OUTLINE" Return
```

and select the location of the note.

---

### Note

If you want a delimiter in an associated text string to separate the associated text from the next piece of information you will enter (usually the component's location), a space must follow the text you want to associate to a component and precede the final quote mark.

---

5. Draw the rectangle defining the active drawing area.

6. Enter:

ADD N :T1010 :AC

and select the active drawing area rectangle.

7. Enter:

"SCREEN"

and select the location of the note.

You are now ready to begin adding parts to the area of the tablet menu inside of the outline rectangle but outside of the active drawing area rectangle.

8. Place rectangles in the drawing for each part you want to put on the tablet menu.

9. Enter:

ADD N :T1010 :AC

and select the rectangle that surrounds a part you want to add. If you want to display the associated text on your plotted tablet menu, you may add :D to the command line shown above. The default is to not display associated text.

10. Enter the text string you want to associate to the part. For example, if your part is called `nand2`, you could enter:

"nand2 "

This text string is the information passed to the Graphics Editor whenever you select the menu item. You can include commands and macros in this string if you wish.

11. Select the highlighted rectangle that surrounds the part you will add. If the boundary rectangle of the text is too large or small for the highlighted rectangle, enter :F and a font size that enlarges or shrinks the boundary rectangle to the correct size.

12. If you plan to add a part from a directory or volume not listed in the current volume search table, add this directory or volume name to the search table (by specifying `SEARCH volume name: ;`).

13. Enter ADD and the part name. If the part was created in another personality the system may ask you if you want to change the units. Enter  for No.

14. Place the part on the tablet menu with your stylus. If the part is too large or small for a particular area, enter :X or :S and a scale factor that enlarges or shrinks the part to the correct size.

The associated text supplies the information the system requires to enact the menu item each time you select it. In the above example, the associated text specifies to the system that `nand2` is to be added after you enter ADD and select `nand2` from the tablet menu.

A handy macro for adding associated text is:

```
DEF TMENU_BOX
  {'Select the rectangle' X;Y;
   'Enter the text ("XXXXX") and the XY Placement Point'}
  "ADD N :T1010 :AC {X},{Y} ";
```

This macro adds a note component on the default layer (since no layer is specified), assigns it the tag value 1010, and associates it to the rectangle located at X;Y. Once you have defined this macro, you may enter TMENU\_BOX each time you need to associate text to a rectangle surrounding a menu item. The screen prompts tell you when to enter the text string and placement point. If you want to toggle the display of the associated text on and off, enter WIN :A;.

15. Repeat steps 7 through 14 for each part you wish to add to the tablet menu.

16. Enter:

```
SAV SEARCH; Return
```

to save the volume search table in current memory. This is necessary if you modified the current search table before you added any parts to the tablet menu.

17. Enter

```
SAVE
```

and the name of the tablet menu. The system will append the suffix .d to the file name.

## Plotting a Tablet Menu

You can plot the tablet menu on either paper or an acetate overlay. Before plotting, the tablet menu must be displayed on the screen.

### On Paper

If you are using paper, load the plotter with C size or greater paper. A fine tip pen (i.e., 3 mm) is suggested.

Enter:

```
PLO :C :S1 :V20;
```

for a local plotter. If your plotter is connected to an SRM, you must also enter the SRM plotter directory path, a file name, and paper size. Make sure the scale and paper size values in this command line are appropriate for your specific drawing. The command shown here assumes the tablet menu drawing is to be plotted at a scale of 1.

### On a Pre-cut Overlay

If you are using a pre-cut acetate overlay made for your graphics tablet, enter:

```
WIN :F;  
WIN X1,Y1 X2,Y2;
```

X1,Y1 is the lower left and X2,Y2 is the upper right corner of the drawing. The drawing should completely fill the display screen.

Place the blank acetate overlay on the graphics tablet and mark the lower left corner of the entire drawing area (as outlined on the tablet). A fine tip pen (i.e., 3 mm) is suggested.

Place the acetate on the plotter and enter:

```
PLD :S1;
```

The system will ask you to specify P1 on the plotter. P1 is the point (lower left corner) you previously marked on the acetate.

Press **Return** to continue plotting. (For most plotting operations you need to specify both P1 and P2; however, when plotting a tablet menu HP EGS will ignore P2.)

### Activating a Tablet Menu

The SAVE TMENU command creates a special tablet menu file which contains the tablet data. To activate the tablet menu:

1. Place the cut-out paper or acetate tablet menu on the graphics tablet.
2. Enter:

```
SAV TMENU filename ;
```

The file name must be less than six characters. The system will append the suffix `_tm,TEXT` to the file name.

3. The system prompts you to digitize the four corners of the tablet menu . You must select the corners in the order asked for by the system.
4. Once you have selected the corners, the tablet menu becomes active and remains active until you leave the Graphics Editor, load a different tablet menu, or toggle the menu off with `TMENU ;`.

To add any of the items displayed on the tablet menu to your drawing, select `ADD*` from the screen menu (or enter `ADD` from the keyboard) and then select the tablet menu item. However, if `ADD` is part of your associated text string, you need only select the tablet menu item and the desired location on the screen.

To turn off the tablet menu, enter:

```
TMENU ; Return
```

This command will not work if your tablet menu is not loaded in the computer's current memory. If this is the case, simply use the LOAD command (i.e., LOAD TMENU filename ;) to bring the tablet menu into memory. Then toggle the menu on or off with the TMENU command.

## How the System Interprets Menu Items

When you select a menu item from a tablet menu, the system will interpret the menu item information in one of two ways.

- If the menu item is a system command, the Graphics Editor understands and performs the command. This is also true for user-defined macros. Commands and user-defined macros are case insensitive.
- If the menu item contains other information, the system reads and acts on the information contained in the associated text string just as if you had entered it on the keyboard. This information is case sensitive.

In the previous section, the example's associated text string was "nand2 ". Notice that the example has a space between nand2 and the last set of quotes. The system interprets this space like **Return**. For example, numbers are tablet menu items that should not have a space inserted between the associated text and the second quotes. This is because you may want to combine several numbers together (i.e., 234), and if each ended with a space (i.e., 2 3 4), the system would always enter each number as soon as you selected it. Without a space, the system waits for **Return**, a semicolon, or ENT (on the screen menu) before it enters a number string.

## Space Planning Example

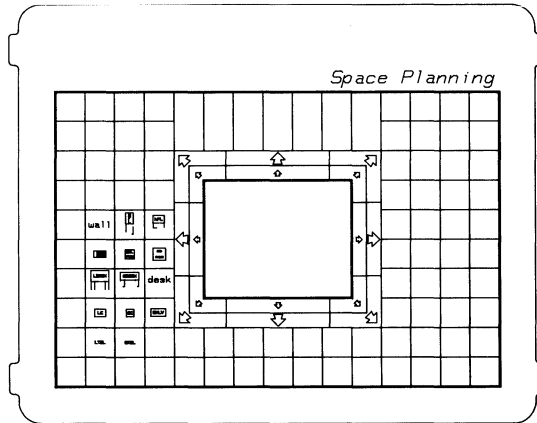
The drawing defining the tablet menu and the tablet menu file for the example Space Planning personality are:

```
EWSP:sptmenu_d
EWSP:sp_tm,TEXT
```

You can view the tablet menu drawing by calling up the Space Planning personality. Then enter:

```
edi EWSP:sptmenu;
win :n10;
```

The WINDOW command lets you see all the detail of the parts. The Space Planning tablet menu looks like this:



The tablet menu was constructed using the procedure described in the previous section on drawing tablet menus. Briefly summarized, the General Drawing tablet menu served as our template. Library parts were added to the menu using the ADD *<instance>* command (for example, ADD EWSP:1desk). If the library part was the wrong size for a particular menu box, the part was scaled up or down with the :X option.

The ADD N command associated text to each menu item. For example, in one case we entered ADD N :T1010 :AC , selected the menu item with the stylus, entered "1desk " , and placed the text string in the box enclosing the menu item. If the text string was the wrong size for a particular menu box, we scaled the text up or down with the :F option.

After the tablet menu was completed, we saved it by entering SAVE EWSP:sp\_tmenu;. The SAVE TMENU command creates a text file which contains the coordinates of the tablet menu corners and each item contained on the tablet menu. The contents of the completed Space Planning tablet menu file appear in EWSP:sp\_tm.TEXT.

You can plot the Space Planning tablet menu onto C size paper, cut it out, and place it on your graphics tablet. To activate the menu, enter SAVE TMENU EWSP:sp; and select the four corners in the order asked for by the system.



## Modifying the Message Files

HP EGS supplies message files for the Graphics Editor and for each personality. These files contain the initial messages that appear when you first begin using HP EGS, as well as error messages and HELP messages for Graphics Editor commands and personality command macros. You may want to change these messages to read differently or translate them into other languages.

A message file is a specially organized ASCII file that permits application programs to retrieve information quickly and conveniently. Message files end with the suffix `_ms.ASC` and are created and modified with the Pascal Editor.

### Structure of a Message File

A message file groups logically related sets of information in a single location. Each set of information in the file is denoted by a “tag field” which names the information in the set and provides access to it. A message file consists of one or more sets of information where each set of information takes the following form:

```
tag field ^D^ information
```

The `tag field` is the name of the information field. It is limited to 20 characters. In most cases this specifier is fixed and must not be altered by the user. Each tag field in a file must be unique. Leading and trailing blanks are not significant.

The `^D^` characters are delimiters.

`information` is the information related to the specific tag field. It may consist of any number of lines of textual information, with no line exceeding 255 characters. The use of this information is defined by the application program. Only the first line of information (the line containing the tag field) need contain a flag (`^D^`), although the flag may be present in additional lines.

Two special message tags appear in the file `EWCODE:gedit_ms.ASC`. They are `macro_enter` and `macro_eoc`. These tags substitute for either a `<cr>` or a semicolon. You may change the values of these tags (`ENT` and `EOC`) if desired, particularly for foreign languages.

### Runtime Parameters

Many of the messages displayed by the HP EGS system contain runtime parameters. A string of characters of the form `{# number description}` denotes a runtime parameter. For example, one of the error messages for macros is:

```
macro_mod291 ^D^The macro name '#{1 name}' is too long (MAX = #{2})
```

When this message is displayed, the string `{#1 name}` is replaced by the name that you entered and the string `{#2}` is replaced by a number. When no description is included with the runtime field, the runtime parameter is a number (maximum, minimum, error number, etc.). Should you modify a message containing runtime parameters, include the runtime fields in the modified message. For example, you might change the message shown above to:

```
macro_mod291 ^D^The limit of {#2} characters for a macro name
             ^D^was exceeded for macro name '{#1 name}'
```

Here the single line message has been broken into two lines and the order of the runtime parameters reversed. This is allowable as long as one does not violate the rules given above. Had the user changed the message to:

```
macro_mod291 ^D^Macro name too long
```

the program would still function correctly, but the information about the name and maximum size would not be displayed.

## Editing and Indexing a Message File

Message files are modified with the Pascal Editor. After you have modified the file, it is a good idea to check it to make sure that you have not corrupted the file. A utility program `EWSYS:testmfile` can be used to check a message file and display a list of any errors encountered. In addition, you can use this utility program to generate an index to a message file.

An index to a message file contains a list of the tags in the related message file and an index to the location of the beginning of each message. This file is stored with the suffix `_mi,ASC`. With short message files (such as the Manager's control file) no index file exists; rather, the index is built in memory each time the program is run. However, for a large message file (such as `EWCODE:gedit_ms,ASC`), it can take several minutes to generate an index each time the file is loaded. With an index file, it takes only a few seconds to load.

Every time you modify a message file (i.e., after adding a new message), you must "compile" or re-generate its index. This index file is stored outside the Graphics Editor.

To check a message file you have modified and generate an index to the message file, you need to do the following:

1. Select the System Utilities menu.
2. Select the Message File Utility from this menu.
3. When the program prompts you for the name of the message file, type in the complete file specification (i.e., `EWCODE:gedit_ms,ASC` or a full SRM path name). Be sure to include the suffix `,ASC` with the file name.

4. The program will check the message file and report any errors on the alpha display. The errors could include such things as tag values exceeding 20 characters, duplicate tags, the information field being too long, or the file entered not being an ASCII file. At this time you should fix any errors and re-generate the message file index.
5. Next the program will ask you for the name of the new index file. Enter the complete file specification with the suffix `_mi,ASC` (i.e., `EWSYS:gedit_mi,ASC` or a full SRM path name). Note that this file must reside in the volume to which you are prefixed when you load HP EGS (`EWSYS:`). Press **Return** and the system will store the new index file on your disc. The index message file is now ready for use.
6. The program also will prompt you to enter a message tag (i.e., `PLOTTER`). If you enter a valid tag, the program will display the message associated with it. If you press **Return** without keying in a tag, the program terminates. If you press **?**, the program lists all the tags in the file.

## Examples of Message Files

HP EGS uses the following message files to display prompts and error messages:

```
EWCODE:gedit_ms,ASC
EWME:me_ms,ASC
EWEE:pc_ms,ASC
EWEE:sch_ms,ASC
EWOPT:toiges_ms,ASC
EWOPT:toegs_ms,ASC
EWOPT:phdr_ms,ASC
EWRC:con_ms,ASC
```

Note that all of the files end with the suffix `_ms,ASC`. These files are normal message files with these additional limitations:

- No message displayed in graphics area may exceed more than 255 characters.
- No message displayed in the graphics area may contain more than six lines.
- If the first character of a message is `@`, the program will beep when the message is displayed in the graphics area.

Let's look at some examples from the Graphics Editor message file `EWCODE:gedit_ms,ASC`. Suppose we want to send a drawing to a plotter. HP EGS needs to know the select code and bus address of various peripherals in order to access them. This information is stored in the message file using "field specifiers" or names associated with the various peripherals. These names (the tag fields) are defined by HP EGS and must not be changed. For example, the tag field `PLOTTER` is used to access the graphics output device. However the information field containing the select code and bus address information may be changed if these parameters are different. The default entry for the graphics output device is:

```
PLOTTER^D^705
```

This defines a message whose tag field is PLOTTER and whose information is the string 705. Recommended values for peripheral select codes and bus addresses may be found in *Installing HP EGS*.

Each prompt or error message displayed by HP EGS is stored in the same message file. This allows others to change the message file to show alternate text. This is most often done to provide foreign language prompts. For example, one entry in the message file might be:

```
no_Plotter ^D^ No Plotter available
```

In a French version of the software, the same message might be:

```
no_Plotter ^D^ Pas de table tracante disponible
```

whereas, in German, the message might be:

```
no_Plotter ^D^ Keine Plotter angeschlossen
```

Note that in both examples, only the information itself was changed; the tag field was not changed. Thus, any application program can be easily customized by simply changing the information section of each field.

## Help Messages for Personalities

Because most personalities are built using macros which you define, it is desirable to have some form of help for the user of a personality. This help is in the same form as message files which have been just described.

When you have finished your personality and all of the associated macros, you may want to describe these macros and give help for them. To do this, simply build a message file containing tag values named the same as the macros for your personality. Then include the command LOAD HELP <file specification> in your start file. Whenever anyone uses the personality, your help file will be loaded into the Graphics Editor along with the system error messages and prompts. Then if the user requests help for a macro you have defined, the system will get the text from the help file you have created and display it.

This technique allows you to make your personality easier to use and friendlier for the novice user. You can look at the help message files for the HP EGS personalities for more examples and to gain a better understanding of this concept. They appear in the following files:

EWYS:gd_ms,ASC	General Drawing Help File
EWEE:pc_ms,ASC	PC Board Layout Help File
EWEE:sch_ms,ASC	Schematic Drawing Help File
EWME:me_ms,ASC	Mechanical Drafting Help File

## Creating New Stroke Tables

A stroke table is a file that defines the strokes used to draw all notes and text. The HP EGS Graphics Editor uses two stroke tables: one that defines note characters and another that defines text characters. By modifying these files or by loading other stroke tables, you can change the characters drawn by the system.

Stroke tables are text files stored with the suffix `_st.TEXT`. The two default files are `EWSYS:note_st.TEXT` and `EWSYS:text_st.TEXT`. You may modify these files as needed using the Pascal Editor. Several additional stroke tables listed below are also included with the system.

### HP EGS Stroke Files

Stroke File	Description
<code>EWSYS:note_st.TEXT</code>	Note characters
<code>EWSYS:text_st.TEXT</code>	Text characters
<code>EWSYS:kata_st.TEXT</code>	Katakana (Japanese) note characters
<code>EWCODE:note45_st.TEXT</code>	Note characters compatible with EGS/45
<code>EWCODE:text45_st.TEXT</code>	Text characters compatible with EGS/45
<code>EWCODE:smooth_st.TEXT</code>	Smooth text without proportional spacing
<code>EWCODE:roman_st.TEXT</code>	Roman text without proportional spacing
<code>EWCODE:smthPP_st.TEXT</code>	Smooth text with proportional spacing
<code>EWCODE:romPP_st.TEXT</code>	Roman text with proportional spacing

If you create your own stroke tables, give them the suffix `_st.TEXT` so that the Graphics Editor recognizes them. Then either replace the default files with your own files or use the `LOAD NSTROKE` and `LOAD TSTROKE` commands to load your files while the Graphics Editor is running. Note that you may load only one note and one text file into the Graphics Editor at any time.

### The Stroke Table Format

The format of a stroke file is as follows:

- The first line of the file specifies the size of the character cell and an optional slant value. The character cell may be as large as 127 by 127, although the standard HP EGS tables use a 16-by-16 cell. The larger the cell, the smoother the character may be. The actual height of the character is controlled by the font specified when the text is added to a drawing. The text slant is expressed in degrees and may range from  $-75$  to  $+75$  degrees. If no slant is specified, the system assumes a slant of 0.
- The second line of the file lists the fill instructions. If the text string `FILL ALLOWED` appears on the second line, you can fill text. Recall that only text characters may be filled in HP EGS. Therefore, if you define your own stroke table and wish to fill characters, you must load your file as a text stroke table. If any other string appears on the second line, fill is not allowed.

- The succeeding lines of the file define individual characters and the strokes that make them up. The sequence that defines the character begins with the character itself. You may define characters with either a key shown on the keyboard or with ASCII ordinals in the range 0 through 255. We recommend that you avoid defining characters with ordinals in the ranges 0 through 31, 128 through 142, and 255, as these ordinals specify special characters. Of course, it is not necessary to define every possible character in the file. To specify a new character, press **Shift** **/5** (the **ANYCHAR** softkey) and the selected three-digit number, or press the selected key on your keyboard.
- Following the character you must list the coordinates that define your special character. These coordinates are expressed in terms of the coordinates of the character cell defined in the first line of the file. As HP EGS interprets these coordinates, it draws from the first point in the list to each successive coordinate pair. If it is necessary to lift the pen to begin a new stroke, insert the coordinate pair -128 0 in the sequence. The entire sequence must end with the coordinate pair -128 -128. The entire list of coordinates may appear on more than one line, and no commas should be used in the file.

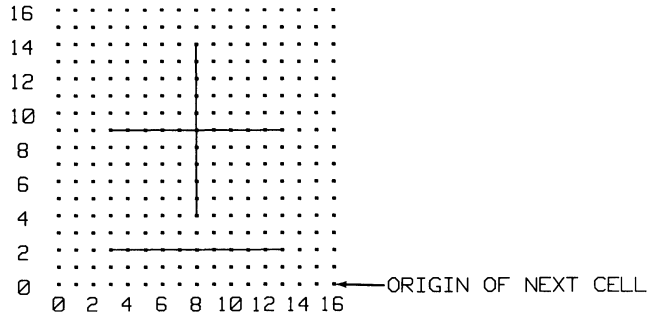
To indicate you want proportionally spaced characters, place a single value after the terminating coordinates. This value represents the character cell x-axis coordinate where you want your next character to start. If no value appears after the terminating coordinates, the system assumes the next character should begin where the previous cell ended. This value must appear on the same line as the terminating coordinates.

## Example

Suppose we want to assign the plus-minus sign (which does not appear anywhere on the keyboard) to a key on the keyboard. To assign this shape to the ASCII ordinal 160, given a 16-by-16 cell, we could type the following lines in the appropriate stroke table file:

```
16 16
FILL NOT ALLOWED
h_p 8 4 8 14 -128 0
      3 9 13 9 -128 0
      3 2 13 2 -128 -128
```

This file would create a drawing that looks like this (without the grid guidelines):



Thus, when we are working on a drawing and we press **Shift** **f5** (the **ANYCHAR** softkey) 160 on our keyboard,  $\pm$  appears in our alpha display and a drawing of a plus-minus sign appears in our drawing. The line type that appears in the drawing is controlled by the layer on which we add the stroke, and the size of the stroke is controlled by the font size we select at the time we add the stroke. The width of this stroke is 16 cell coordinates wide, since we did not specify any other desired x-axis endpoint after the terminating coordinates. Had we specified 14 after the terminating coordinates, the next character's origin would have been at the 14,0 point of this cell.

## Troubleshooting

If you cannot successfully load a stroke table you have modified or created, check the following things:

- Is there anything after the last -128 -128 in the entire file? To determine if there is, use the Jump End function of the Pascal Editor. The cursor should be positioned immediately following the last 128. Similarly, use the Jump Beginning function to check if the first line defines the cell size.
- Is there anything after the terminating -128 -128 for each character? If you have not defined a proportional spacing value, place the cursor on the right end of the last -128. Pressing the right arrow key once should move the cursor one character beyond the 8. A second push should move it to the next line. If it does not, delete the excess blanks at the end of the line. If you have specified a proportional spacing value, it must be the last thing on the line. Use the same technique to verify that no spaces follow that value.
- Does each character definition contain pairs of coordinates?
- Are the coordinate values in the range 0 to 127 and within the defined cell size?

## Putting the Personality Together

At last, you are ready to use your new personality! As a final check of your work, make sure you have created or modified all the files necessary to run the new personality. Take a look at the notes you made when you first defined the personality. Does your personality include all the features you said it should?

As you execute HP EGS and call up your personality, pay close attention to the screen messages you see. Check to see if all your files are loaded. If loading fails, read the alpha display to pinpoint where the loading procedure stopped. Then call up the problem file(s) and check for syntax, formatting, and naming errors.

Once the personality is loaded, test each menu item to ensure that it works. If you select an item and it does not produce the desired result, check the following:

- Macro file: Is this file loaded? Is the name of the menu item you selected exactly the same as the macro name in the macro file? Does your selected macro name duplicate the name of a Graphics Editor command, another macro, a library part, or a layer label (in the process file)?
- Tablet menu file: Is your tablet menu activated? You may toggle this menu on and off by typing `TMENU;` .
- Volume search files: Have you listed all the volumes needed to make your menu items work in the Manager's and personality's volume search tables? Are all the volumes on line?

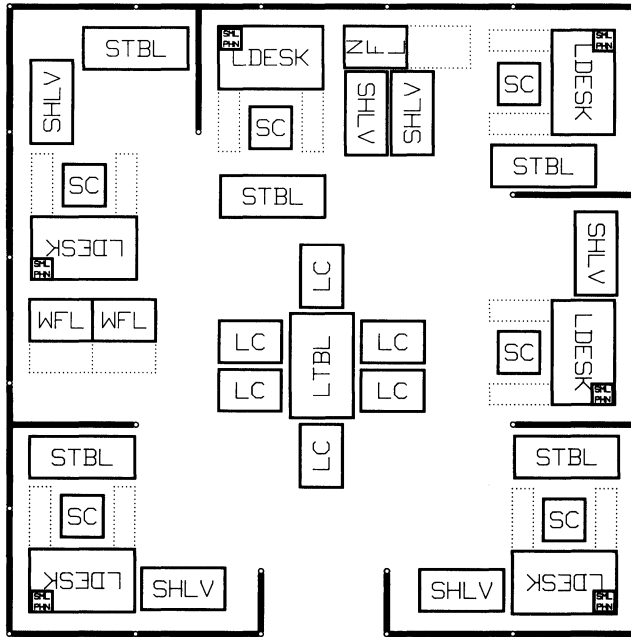
Next check the functionality of your personality. Does it do what it was designed to do? Test it by making some actual drawings and using your own application programs (if any exist). If any features do not work as planned, refer back to your original personality definition to confirm that you need the features in question. Then revise the files that affect that functionality.

### Space Planning Example

To load the entire Space Planning personality, we must first enter HP EGS. The first menu we see on the screen should contain the Space Planning option. If we select this option, the system will execute `edit, CODE` and begin loading all the personality's files.



Once all the files are loaded, the system will give us the message The INPUT file has finished. We are now ready to draw a floor plan, like the one pictured here:



To figure square footage used in various parts of the drawing, we can use the AREA and DISTANCE commands. We can also use the SHOW command to show selected items in the drawing, and experiment with the OPEN\* and CLOSE\* macros to show desk drawers and file cabinet doors both opened and closed. When we are satisfied with the drawing that appears on the screen, we can save the drawing and plot it.



# Material Listing

## Chapter 3: Material Listing

Introduction .....	103
The LIST MATERIAL Command.....	104
Drawing Conventions .....	104
LIST MATERIAL Output File.....	104
The Parts File Editor .....	105
Selecting a Parts File .....	105
Parts File Structure.....	105
Modifying a Parts File .....	108
Listing a Parts File .....	109
The Electrical Engineering Material List Post-Processor .....	110
Generating a Post-Processed Material List .....	112
Material List Error Checking .....	114
The Mechanical Engineering Material List Post-Processor .....	115
Material List Output Format .....	117
Generating a Post-Processed Material List .....	120
Listing a Post-Processed Material List .....	122
Material List Error Checking.....	123



# Material Listing

Chapter

3

## Introduction

This chapter describes how to get material lists from your HP EGS drawings. Before reading this chapter you should understand how to use associated text and the LIST MATERIAL command. More information on these topics is in Chapter 1: *The Graphics Editor*, and in the sections describing the ADD TEXT/NOTE and LIST MATERIAL commands in the *HP EGS Syntax Reference*.

HP EGS supplies multiple tools to aid you in creating material lists. These tools include:

- |                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Electrical Engineering<br>Post-Processor    | This post-processor creates a formatted material list designed for the electrical engineer. It counts basic library parts (dips, and gates, etc.), and understands concepts like reference designators.                                                                                                                                                                                                                                      |
| Mechanical<br>Engineering<br>Post-Processor | This post-processor creates a formatted material list designed for the mechanical engineer. Instead of counting library parts, like the Electrical Engineering post-processor, it counts annotated “bubbles” you have placed around items.                                                                                                                                                                                                   |
| LIST MATERIAL<br>Command                    | This command may be executed from the Graphics Editor. From all the parts contained in a drawing file, it creates an ASCII text file which is human-readable and editable. The two post-processors mentioned above read the LIST MATERIAL output file, and combine that information with a parts file to create the final formatted material list. See the <i>HP EGS Syntax Reference</i> for more information on the LIST MATERIAL command. |
| Material List<br>Parts File Editor          | Both the Electrical and Mechanical Engineering post-processors use a common parts file editor, although the parts files they use have different content. The purpose of the parts files is to contain text information that you do not want cluttering each part in your drawing. For example, having a 50-character description on each part would waste drawing space.                                                                     |
| Personalities                               | The HP EGS personalities have menus and macros to aid you in making drawings that meet the conventions for material listing discussed in this chapter.                                                                                                                                                                                                                                                                                       |

In general, the post-processors take information from the LIST MATERIAL output file and build keys into the parts files to get the rest of the information (description, stock number, etc.) about each part.

## The LIST MATERIAL Command

LIST MATERIAL is a Graphics Editor command that inputs a drawing (if it is not already in memory) and outputs a data file from that drawing. Using either the Electrical Engineering or Mechanical Engineering post-processor, you may combine this data file with a parts file to produce a post-processed material list.

### Drawing Conventions

To create the correct output from the LIST MATERIAL command for one of the post-processors, you must follow certain conventions when you make drawings. You must associate text with certain tag values to components in the drawing (using the ADD TEXT/NOTE command). This information is later used as a key into a parts file and/or included verbatim in the final post-processed material list.

For more information on adding associated text, please refer to the ADD TEXT/NOTE command in the *HP EGS Syntax Reference*. The MOVE command can be used to change the location of the associated text, and the MODIFY command can be used to change the display mode (displayed vs. non-displayed), the tag values, and the contents of the text.

The tag values for the Electrical Engineering post-processor are:

1001	Reference designator
1004	Part type, value, or model name

The tag values for the Mechanical Engineering post-processor are:

1051	Item number
1052	Quantity
1053	Identifying string (Part or drawing number)
1054	Size
1055	Zone

### LIST MATERIAL Output File

The delimiter used to separate the different columns in a material list output file is the vertical bar |, which is ASCII character 124. This character is encoded in the message file so that you can change it if you desire. However, if you change the character, be sure to change it in all message files: EWCODE:gedit\_ms.ASC, EWME:me1\_ms.ASC, and EWEE:matlst\_ms.ASC.

The format of the LIST MATERIAL file is:

```
component | tag text 1 | tag text 2 |...
component | tag text 1 | tag text 2 |...
component | tag text 1 | tag text 2 |...
```

The first column in the LIST MATERIAL output is a fixed length of eight characters. The valid entries for this first column are the component descriptors A, C, D, L, M, N, O, P, R, T or a library part name. There are no trailing blanks in associated text, and all associated text for the same tag value is printed. If a specified piece of associated text is missing the column will be empty. All files created with the LIST MATERIAL command end with the suffix `_m`.

## The Parts File Editor

A parts file contains detailed information, like stock number and description, for each part you want to include on a material list. The parts file editor lets you create or update your parts files. Once the parts files are created they can be printed (with sort options) and used by the HP EGS material list post-processors. All parts files end with the suffix `_pf`.

Parts files are text files which you can edit with the Pascal Editor. This text format also allows you to access parts files programmatically (i.e., to convert an existing parts data base to the HP EGS parts file format).

### Selecting a Parts File

The first thing the parts file editor does is optionally find existing parts files according to a specified search table (the name of the HP EGS Manager's volume search table is used by default). The parts files it finds are then presented to you for selection, along with any parts files on volumes you specify separately. If you create a new file or change an existing parts file, the changes are saved immediately when the change is made. Once you have modified an existing parts file, you may select other parts files for modification or quit the parts file editor.

### Parts File Structure

When creating a new parts file, you can format it to suit your needs. The parts file can contain as many parts as you desire, and each part can have up to 20 fields. Each of these fields can contain a maximum of 255 characters. To facilitate the processing of the parts file, the parts file editor lets you specify the following things:

- Number of fields per part: Integer between 1 and 20.
- Determination of fields to be used as the key: Appropriate integer(s). Any field can be used as a key field, and there can be as many as 5 key fields. When you list a parts file using the parts file editor, it will be sorted alphanumerically by the keys.

- Name of the field: A string less than 255 characters long.
- Maximum length for a field: Integer between 1 and 255.
- Legal characters for a field: A blank entry implies all characters are legal. Otherwise set notation should be used. Examples of set notation are:

a,,z denotes all lower case letters, where ,, means an ASCII character range.

YNyn denotes both uppercase and lowercase y or n (yes or no).

a,,zA,,Z?\* denotes uppercase and lowercase letters, a question mark, and an asterisk.

Xabc,,zA,,WYZ denotes all uppercase and lowercase letters.

These specifications may differ for each parts file, but they may also produce incompatibilities post-processors may not recognize. For this reason, you may want to specify the name of an existing parts file to use as a template for the structure of a new parts file. If you are using one of the HP-supplied post-processors, be careful when changing the templates. We suggest creating a new parts file with the new template, and testing the changed template before you change your original files.

The format of a parts file is:

```

<number of fields per record>
<key field #1> <key field #2> , , , <key field #5>
<Name of field #1>
<Max. length of field #1>
<Legal chars for field #1>
<Name of field #2>
<Max. length of field #2>
<Legal chars for field #2>
.
.
.
<Name of field #n>
<Max. length of field #n>
<Legal chars for field #n>
<-- Blank line between template & record data

<Record 1 field 1>
<Record 1 field 2>
.
.
.
<Record 1 field n>
<-- Blank line between records

```



```

<Record 2 field 1>
<Record 2 field 2>
.
.
.
<Record 2 field n>
.
.
.

```

<-- Blank line at end of file

---

**Note**

A utility is provided to convert EGS/200 1.0 and 1.05 parts files to the new HP EGS 2.0/2.1 parts file format. To use this utility, select the option on the Electrical Engineering menu named `Convert Old Parts File`. This utility will create a copy of the old file in the new format.

---

The specified blank lines shown in the previous example are required, and any extra blank lines are illegal. If desired, you may place comments in a parts file by starting each line of the comment with \$\$.

When specifying the list of key fields (the 2nd line of the file), put the numbers of the five fields in the order you want them sorted. For example, if you wanted the key to be a combination of field 1 and field 3 the second line of the file would look like this:

```
1 3 0 0 0
```

The template for the HP EGS Electrical Engineering parts file (EWEE:eepart\_pf) is:

```

5
1 2 0 0 0
Part Name
8

Part Type
16

List Flag
1
yYnN
Stock Number
11

Description
40

```

The template for the HP EGS Mechanical Engineering parts file (EWME:mepart\_pf) is:

```
4
1 2 0 0 0
Part Number
14

Drawing Number
14

Description
40

Specifications
40
```

## Modifying a Parts File

Three options are available for adding parts to a parts file or modifying an existing file:

- **ADD** a new part: In this mode a part can be added to the selected file. The system prompts for each field value and checks for character and length restrictions. The system then checks the value of the key field against all parts in the file; the new record is rejected if the key is not unique.
- **CHANGE** a part: In this mode any field in an existing part can be changed. The part is accessed by its key. The system prompts you for the value of each key, and the key field values must exactly match the record. Next the system presents all the fields for that part. When you select the field to be changed the screen will be cleared and the current value of the field will be displayed. You may use the keyboard editing keys (**Insert char**), (**Delete char**), etc.) to change the field. Press (**Return**) when you are done changing the field. You may then select another field to be changed, or **QUIT** to exit the change mode.
- **DELETE** a part: In this mode any part can be deleted from the file. The part to be deleted is specified by its key. Each key field value must match the part record.

## Listing a Parts File

A selected parts file can be sorted and then listed to one of three locations. The procedures involved in doing so are as follows:

1. The system asks you where you want to output the list. There are three valid locations:
  - a. **CONSOLE:** This lists the parts file to the computer display. The listing pauses each time the screen is filled.
  - b. **PRINTER:** This lists the parts file to the currently defined local printing device. The number of lines per page is requested for automatic pagination.
  - c. **File:** This lists the parts file to the specified file. Again the number of lines per page is requested, as this feature can be used for SRM printer spooling.
2. Next the system asks if you want to specify the sort key (Yes or No). If you specify No, the system uses the key defined when the file was created (i.e., the second line of the file). If you specify Yes, the system prompts you for a series of key fields. You should enter the number of each desired key field, and then enter 0 when you want to stop defining new keys. Note that if you enter 0 without keying in any new fields, the system will immediately start printing an unsorted file (in the same order as the original file's key fields).
3. The system then orders the file index alphanumerically by key field values.
4. The system prompts you for the value of each key field, and compares the fields you enter to the sorted records.
  - a. If an exact match is found, that record is listed, followed by all successive records in the alphanumerically sorted list.
  - b. If no match is found, the next record beyond the match failure is listed first followed by the alphanumerically sorted list.
5. After the list is completed, the system returns you to the menu screen:

```
ADD
DELETE
CHANGE
LIST
Select new file
```

## The Electrical Engineering Material List Post-Processor

The Electrical Engineering material list post-processor takes part names and part types from a LIST MATERIAL output file. It strings them together to build a key into the parts files, and creates a final output format useful for electrical engineering applications. To understand the Electrical Engineering material list post-processor, it is helpful to know how we define the following terms:

- Part – An HP EGS library part. The part name will be included in the LIST MATERIAL output file and is part of the key into the parts file. The part name may contain 8 characters.
- Part Type – A delimiting term for a part (i.e., the resistor that is type 10k, or the pipe that is 2-inch copper). The part type will be included in the LIST MATERIAL output file and is part of the key into the parts file. The tag value for type is 1004. The type may be 16 characters long.
- List field in parts file – Allows you to indicate whether the part is to be listed in the post-processed material list. An entry in the parts file is required for each library part in your drawing. However, since you may not wish to list every part that appears in your drawing, this option field allows you to choose whether or not to list a part. A title block is a good example of a part you would not want listed.
- Stock # in parts file– Your company's stock number for that part. You select its width at the creation of the parts file. It may be from 8 to 20 characters long. The default length is 11 characters.
- Description in parts file– A brief description of the part.
- Reference designator in LIST MATERIAL output file – A unique name associated to instances which allows the material list to differentiate between multiple occurrences of the same library parts. The reference designator may be up to 250 characters. For stepped instances, the associated text for a reference designator contains a string of references for the entire array. If there are not enough reference designators for every member in the array, the default (\*) is assigned to each of the leftover instances. The tag value for a reference designator is 1001.
- Post-processed material list – A formatted listing of parts produced by the Electrical Engineering material list post-processor.

### LIST MATERIAL Output File

When you execute the Electrical Engineering material list post-processor, the system compares the LIST MATERIAL output file against the parts file to generate a formatted material list. The columns in the LIST MATERIAL output file that the Electrical Engineering post-processor expects are:

Component	This column is automatically provided by HP EGS and contains the component descriptor (A, C, D, L, M, N, O, P, R, T) or library part name). For Electrical Engineering material lists, the component will always be a library part name.
Part Type	The part type associated to the library part.
Reference Designator	The reference designator associated to the library part.

To create a LIST MATERIAL output file for the Electrical Engineering post-processor you must do the following:

1. Lay out your drawing with the Graphics Editor, making sure to comply with the following conventions. Each part you want counted must:
  - a. Be saved as a library part at logical level 0. You set the logical level by executing a 'LEVEL 0;' command before saving the part.
  - b. Have a unique reference designator associated to it.
  - c. Have a part type associated to it.
2. Make sure that for each unique combination of a library part name and a part type in your drawing, a corresponding entry exists in a parts file. For example, a resistor having the part type 10K defines a unique part: a 10K ohm resistor. While your drawing may use many instances of the 10K resistor, only one entry in a parts file is required.
3. Use the LIST MATERIAL command to create an output file. The exact command should look like this:

```
LIST MAT -E #I #N52 #N53 :D0 :T[1004, 1001] <drawing name> <output volume>;
```

This command will create a LIST MATERIAL output file from the drawing <drawing name> on <output volume>. The only components that will be output are library parts (#I) on logical level 0 (:D0) and associated notes on layers 52 and 53 (#N52 #N53) with tag values 1004 and 1001 (:T[1004, 1001]). Layer 52 is for reference designators and layer 53 is for part types. To aid you in creating this LIST MATERIAL file, a macro named LIST\_MAT has been provided in the Electrical Engineering personalities with the above definition.

An example of a LIST MATERIAL file for the Electrical Engineering post-processor is:

```
nand2    !74LS00!U3A!  
inv      !74LS04!U1A!  
and2     !74LS08!U2B!  
and2     !74LS08!U2A!
```

## Generating a Post-Processed Material List

To use the Electrical Engineering material list post-processor to create a formatted material list, you must have:

- A parts file which contains the detailed information for each part you wish to include on a material list. This file contains: part name, part type, list flag, stock number, and description.
- A LIST MATERIAL output file based on an HP EGS drawing. Each part in your drawing must be of a type listed in your parts file, and must have an associated reference designator and a part type.

Once you have created your parts file and used the LIST MATERIAL command to create an output file from your drawing, you can generate a formatted material list as follows:

1. Select the Electrical Engineering option from the main Manager menu, and then select the EE Material Lister from the next menu.
2. The system will ask you to enter the volumes to be searched for parts files. Then it will ask you if you want to search the system volumes (EWSYS:, EWCODE:, EWEE:, EWME:, EWOPT:) for parts files. The list of parts file names will be displayed, and the menu will read:

```
RETURN to Material List Menu  
USE ALL Parts Files  
{list of parts files}  
RESTART Selecting Parts Files  
DONE Selecting Parts Files
```

You must tell the system which parts files contain the entries corresponding to your drawing. To indicate which files to use, select each parts file from the set of choices on the CRT. Once a file has been selected, an asterisk \* will appear to the left of the file name. If you make a mistake and select a wrong parts file, simply re-select that file and the asterisk \* will disappear. To select all parts files listed, pick USE ALL Parts Files. To start the selection process over again, pick RESTART Selecting Parts Files. When you are finished selecting the parts files, pick DONE Selecting Parts Files.

3. The system then asks you to enter the names of the material list output files to be included in the post-processed material list. Enter the file names. A maximum of 20 files can be entered. Pressing **Return** without a file name terminates input of file names.
4. The system then asks you whether you want the material list sorted by reference designator or by stock number. Responding **R** causes the list to be sorted by Reference designator. Responding **S** causes the list to be sorted by Stock number. Pressing **Return** causes the system to prompt you for the name of the material list output file.

---

#### Note

Although this may sound contradictory to what you have read so far, reference designators are *not* required for the Electrical Engineering post-processor. Thus, this processor can be used for other types of material lists that don't understand the concept of a reference designator. If all reference designators are missing in an output file, the material list can only be sorted by stock number, and the reference designator column will be omitted from the output.

---

5. The system then asks you to enter the output destination for the listing: **C** for the CRT, **P** for local printer, and **F** for file. If you enter **F**, the system asks for the file name. The system then prompts you for the number of lines per page that you want. The number you supply includes the space occupied by titles and headers. The default number for the printer and for a file is 60.
6. The next system prompt allows you to enter a one line (80 character maximum) page header for your material list. If you press **Return** without entering any characters, no header is printed at the top of the material list.
7. If the material list is not listed on the CRT, the system asks how many copies you want. The default value is one copy.

The resulting material list reports reference designators as follows:

- Reference designators for stepped instances are noted in three ways:

R[1:2] means two components: R1 and R2.

U11[A:D] means four subcomponents within one physical unit U11, which is equivalent to U11A, U11B, U11C, and U11D in non-stepped instances.

C1,C2 means two components: C1 and C2.

- Unit numbers of reference designators are sorted in numeric order. For example, U1,U12A,B,U13A-D,U2 would become U1,U2,U12A,B,U13A-D.
- Reference designators for the same part type and stock number are combined. For instance, suppose we have two library parts: one named nand2, with reference designator U1A, of type 74LS00, stock number 1802-1112, and the other named not\_or2, with reference designator U1B, of type 74LS00, stock number 1802-1112. In the post-processed material list, the reference designators U1A and U1B would become U1A,B for stock number 1802-1112 (sorted by stock number) and for reference designator U1 (sorted by reference designator).
- The delimiters (commas and hyphens) for distinct references (as in U1, U2), sub-units (as in U1A,B) and continuous sub-units (as in U1A-C) are encoded in the message file (EWEE:matlst\_ms.ASC) as delim1, delim2, and delim3.

An example of a material list from an HP EGS parts file (EWEE:eepart\_pf), and the previous example of output from the LIST MATERIAL command sorted by stock number, is:

```

EXAMPLE MATERIAL LIST                                     2-Nov-84 Page 1

STOCK NO      QTY DESCRIPTION                                REFERENCE
-----
1820-1197     1 74LS00 quad 2-input NAND gate      U3A
1820-1199     1 74LS04 hex INVERTER                 U1A
1820-1201     1 74LS08 quad 2-input AND gate       U2A,B

```

## Material List Error Checking

A part missing a reference designator or a part type causes special characters to appear in one or more fields of the material list. These characters indicate the following:

- An asterisk \* indicates a value is missing.
- A blank field indicates that a value is missing or that no corresponding entry in the parts file exists for that part. For example, if you use a library part to represent a ground symbol in a schematic drawing, and if an entry for this part does not exist in a parts file, an error is generated and a blank is listed in one or more fields of the output. You may intentionally leave a field blank in a parts file if you do not want to include a part in the material list.
- Parentheses ( ) indicate duplicate reference designators in the material list. When a material list is sorted by reference designator, any repetition of a reference designator in the list is enclosed in parentheses. When a material list is sorted by stock number, only duplicate reference designators assigned to the same part are enclosed in parentheses.



## The Mechanical Engineering Material List Post-Processor

The Mechanical Engineering material list post-processor takes the identifying string from the LIST MATERIAL output file, and uses it as a key into the parts files. This identifying string must match either part number or drawing number. The data from the LIST MATERIAL output file and the parts file is combined into a format useful for mechanical engineering applications.

The Mechanical Engineering material list post-processor is unique because it is not based on counting HP EGS library parts. Instead it counts any component which has the appropriate associated text. This allows any geometry as well as any library part to be counted. This can be especially useful in the mechanical drawing area where multiple views of parts and smashed parts must be handled.

This post-processor gives you considerable flexibility in determining the output format. You can specify field or column size, column headings, and the order in which items appear in the material list. In addition, you can send the output to a printer or create an HP EGS input file with it. The latter option lets you place the resulting material list directly on a drawing.

To understand the Mechanical Engineering material list post-processor, it is helpful to know how we define the following terms:

- Identifying string in the LIST MATERIAL output file – This string is matched to either part number or drawing number in the parts file.
- FORMAT – A description of a final output format in EWME:meL.ms.ASC
- Post-processed material list – A formatted listing of parts produced by the Mechanical Engineering material list post-processor.

### LIST MATERIAL Output File

The Mechanical Engineering material list post-processor recognizes LIST MATERIAL output files which have at least five columns. Additional columns can be included in the final output by adding more associated text with its own special tag value in the LIST MATERIAL output file, and by having a FORMAT instruction that includes that column in the message file EWME:meL.ms.ASC. Columns can also be added with data from the parts file, again by having a FORMAT that includes that field. The first five columns that the post-processor expects are:

Component	This column is automatically provided by HP EGS and contains the component descriptor (A, C, D, L, M, N, O, P, R, T) or library part name.
Item Number	This column contains the identifying string for the part. The Mechanical Engineering material list post-processor counts items. If the string “drawid” is found in this column the Mechanical Engineering material list post-processor puts the following three fields in the header of the parts list.
Identifying String	This string is compared against each key field in each part in the parts file until a match is found. If multiple parts with the same item number are present in the LIST MATERIAL output file, only one part will need an identifying string. If the part is not found in the parts file, *NF* (Not Found) is put in the undetermined fields of the material list (the ones that would have come from the parts file). This string is used as a part or drawing number.
Quantity String	If this field is left blank, or no integer is found, a quantity of one is assumed for the item. If the field contains a string, the first integer value determines the quantity (i.e., 16 PLACES sets the quantity to 16). A zero quantity is set if the text string REF is found (i.e., 16 PLACES REFERENCE would set the quantity to 0). The value of this text string (“REF”) is specified in the message file EWME:meLms.ASC.
Size String	This field can contain any string or may be left blank. This field is used to distinguish among unique parts which have the same identifying string. If the item number for these unique parts is also the same, the post-processor will separate the different sizes and place them on sequential lines in the material list.

Additional columns in the LIST MATERIAL output file can contain any, or no, strings. These columns can be included as additional columns in the final post-processed material list. Also note that the delimiter for these fields is set in the the Mechanical Engineering material list post-processor’s message file (EWME:meLms.ASC), and that it must match the delimiter provided by the HP EGS LIST MATERIAL command.

To create a LIST MATERIAL output file for the Mechanical Engineering post-processor you must do the following:

1. Lay out your drawing with the Graphics Editor, making sure to comply with the following convention: Each item you want counted must have a component (usually a circle or “bubble”) on layer 8 with the appropriate text associated to it. This text must include the item number and identifying string. The quantity, size, and zone strings are optional.

2. Make sure that for each unique identifying string, a corresponding entry exists in a parts file.
3. Use the LIST MATERIAL command to create an output file. The exact command for the Mechanical Engineering material list post-processor should look like this:

```
LIST MAT -E #I #EB :T[1051,1052,1053,1054,1055] <drawing name> <output volume>;
```

This command will create a LIST MATERIAL output file from the drawing <drawing name> on <output volume>. Components on layer 8 (#EB) and any text associated to those components with tag values 1051-1055 (:T[1051, 1052, 1053, 1054, 1055]) will be output. Layer 8 is where all item information is stored. To aid you in creating this LIST MATERIAL file a macro named LIST\_MAT has been provided in the Mechanical Engineering personality with the above definition.

The following shows an example of a LIST MATERIAL output file:

```
C      | 3 | 4-PLACES |2580-0004|8-32| |
C      | 2 | 2-PLACES |4444-0002|8-32| |
C      | 1 | 2-PLACES |2360-0183|8-32| |
```

The first column tells us that the associated text was associated to circles; this column is ignored by the Mechanical Engineering material list post-processor. The remaining columns, in order, are item number, quantity, identifying string, size, and zone. In this example there was no zone data.

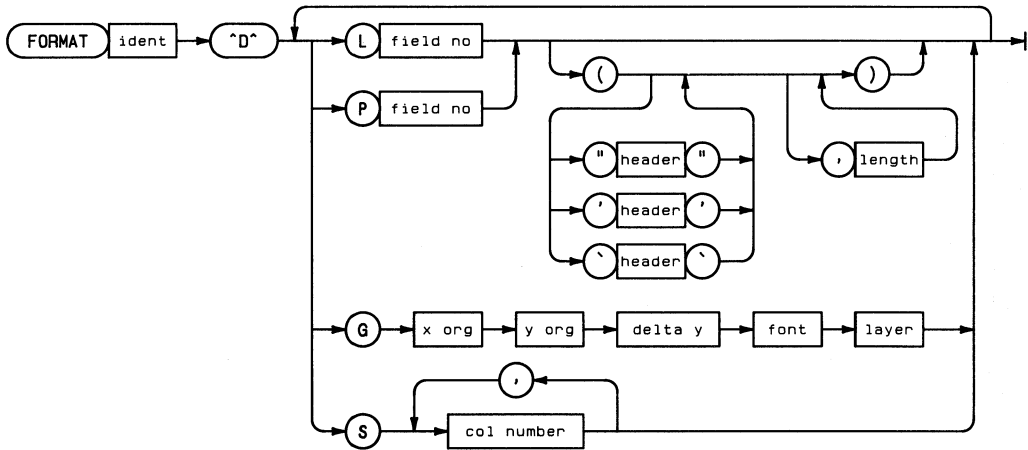
## Material List Output Format

The output of this post-processor can be formatted as you desire. The options include specifying:

- What fields from the parts file and what columns from the LIST MATERIAL output file are to be included.
- The order of the columns in the output.
- The maximum length of each column. If the contents of the column are greater than the specified length, the Mechanical Engineering material list post-processor will create multiple lines for the column to ensure that all the information is included in the final output.

Additionally you can specify that the output be sent to a file for later input to the Graphics Editor. This allows you to place the material list directly on a drawing (using the Graphics Editor's INPUT command).

To specify the desired format, place the FORMAT instructions shown in the following syntax diagram in the Mechanical Engineering material list post-processor's message file (EWME:meL.ms.ASC). You can create as many different formats as desired. Then whenever you use this post-processor, the system will ask you which format you want to use.



Item	Description/Default	Range Restrictions
ident	Unique format identifier. This must be directly appended to the word FORMAT (i.e., FORMATP1).	14 characters
P	Specifies that the content of this column is derived from the parts file.	
L	Specifies that the content of this column is derived from the output of the LIST MATERIAL command.	
G	Indicates that an HP EGS input file containing the material list is to be created.	
S	Indicates the column(s) in the LIST MATERIAL output file to sort on. If the S field is not included, field #2 (item number) will be used for sorting.	

Item	Description/Default	Range Restrictions
field no	Specifies the field to be used. The leftmost column of the LIST MATERIAL output file (component) is 1 and the first (or top) field in the parts file is 1.	Integer
header	This optional descriptor specifies the column heading. If the heading is missing, the following happens: for P, the name of the corresponding field is used as the heading; for L, the number of the column will be used as the heading.	Heading must be in quotes.
length	This optional descriptor specifies the column length. If the length is missing the longest string found for this column determines the length.	Length must be a positive integer.
x org	The x coordinate for the upper left corner of the graphical parts list.	
y org	The y coordinate for the upper left corner of the graphical parts list.	
delta y	The y distance in user units between lines in the graphical parts list.	
font	The font size in user units for the graphical parts list.	
layer	The layer number for the notes in the graphical parts list.	1 - 255
col number	The column of the LIST MATERIAL output file on which to sort. The order of the column numbers specifies the order for the sort.	1 - 5

The following FORMATS are supplied in EWME:meL.ms.ASC:

```

FORMATP  ^D^L2('Item',4)
          P2('Drawing Number',14)
          P1('Part Number',14)
          P3('Description',35)
          L3('Qty',3)
          L6('Zone',4)
FORMATG  ^D^L2('Item',4)
          P2('Drawing Number',14)
          P1('Part Number',14)
          P3('Description',35)
          L3('Qty',3)
          L6('Zone',4)
          G 25,0,0,250,0,125,6
          S 2
FORMATP1 ^D^L2('Item',4)
          P2('Drawing Number',14)
          P1('Part Number',14)
          P3('Description',35)
          L3('Qty',3)
          L5('Size',4)
FORMATG1 ^D^L2('Item',4)
          P2('Drawing Number',14)
          P1('Part Number',14)
          P3('Description',35)
          L3('Qty',3)
          L5('Size',4)
          G 25,0,0,250,0,125,6
          S 2

```

## Generating a Post-Processed Material List

To use the Mechanical Engineering material list post-processor to create a formatted material list, you must have:

- A parts file which contains the detailed information for each part you wish to include on a material list. This file contains: part number, drawing number, description, and specifications.
- A LIST MATERIAL output file based on an HP EGS drawing. Each item you want counted in your drawing must have an identifying string (part number or drawing number) listed in your parts file.

Once you have created your parts file and used the LIST MATERIAL command to create an output file from your drawing, you can generate a formatted Mechanical Engineering material list as follows:

1. Select the Mechanical Engineering option from the main Manager menu. The system then displays:

```
RETURN To Main Menu
Mechanical Drafting Editor
ME Parts File Editor
ME Material Lister
```

2. If you select the ME Material Lister from the menu, the system will ask if you want to search the system volumes (EWSYS:, EWCODE:, EWEE:, EWME:, EWRC:, EWOPT:) for parts files. You will then be asked to enter additional volume names to be searched. Pressing **Return** terminates input of volume names.

The list of parts file names will be displayed, and the menu will read:

```
USE ALL Parts Files
{list of parts files}
RESTART Selecting Parts Files
DONE Selecting Parts Files
```

You must tell the system which parts files contain the entries corresponding to your drawing. To indicate which files to use, select each parts file from the set of choices on the CRT. Once a file has been selected, an asterisk \* will appear to the left the file name. If you make a mistake and select a wrong parts file, simply re-select that file and the asterisk \* will disappear. To select all parts files listed, pick USE ALL Parts Files. To start the selection process over again, pick RESTART Selecting Parts Files. When you are finished selecting the parts files, pick DONE Selecting Parts Files.

3. The system then asks you to select the format to use for the output. This format must be described in the message file EWME:meLms.ASC. Only enter the character(s) that appear after the word FORMAT. For example, if the message name is FORMATP1, you should enter P1.
4. The system asks you to enter the output destination for the listing: CONSOLE: for the CRT, PRINTER: for local printer, or a file name. The system then prompts you for the number of lines per page that you want. The number you supply includes the space occupied by titles and headers.

## 122 Material Listing

5. The system asks you to enter the name of the LIST MATERIAL output file to include in the post-processed material list. Enter the file name.
6. The system asks if you want the post-processed material list in forward or backward order. Enter F for forward, or B for backward. Backward order is typically used for material lists that are to be included in drawings.
7. The next system prompt allows you to enter a one line (80 character maximum) page header for your material list. If you press **Return** without entering any characters, no header is printed at the top of the material list.

The following shows an example of a post-processed Mechanical Engineering material list from an HP EGS parts file (EWME:mepart\_pf), the examples of a LIST MATERIAL output file shown earlier, and the format specified by FORMATP1 (defined in the message file EWME:mel\_ms.ASC):

PAGE 1

MATERIAL LIST EXAMPLE

Item	Drawing Number	Part Number	Description	Qty	Size
1		2360-0183	8-32 Flathead Pozidrive - .750" long	2	8-32
2		4444-0002	8-32 Hexhead Machine Screw - 0.750" long	2	8-32
3		2580-0004	8-32 Hex nut	4	8-32

### Listing a Post-Processed Material List

The Mechanical Engineering material list post-processor lets you send the material list to one of three locations:

- **CONSOLE:** This sends the material list to the computer display. The listing pauses each time the screen is filled.
- **PRINTER:** This sends the material list to the currently defined local printing device. The number of lines per page is requested, and the post-processor includes the header and page number for each page of the parts list.
- **File:** This sends the material list to the specified file. The number of lines per page is requested, as this feature can be used for SRM printer spooling.



When no pagination is requested (by entering '0' as the number of lines per page) no page heading information will be printed. This includes page number, one line header, and any "drawid" information.

For graphical output on a drawing (using the INPUT command), the Mechanical Engineering material list post-processor asks you to specify the direction (forward or backward) to list the parts. If backward is specified, the first part is at the bottom of the list. If forward is specified, the first part is at the top of the list. The post-processor also asks you to specify the name of the file where it should store the HP EGS INPUT commands. This input file will contain commands needed to wrap the parts list and move it, so that you can place the list where you want it in the drawing.

### **Material List Error Checking**

The Mechanical Engineering material list post-processor checks for the following errors:

- Syntax error in FORMAT descriptions in the message file EWME:meL.ms.ASC.
- Missing or illegal fields in the LIST MATERIAL output file.
- Identical items with different identifying strings.
- Identical items with different sizes.
- An item that is in the LIST MATERIAL output file but is not in the parts file (\*NF\*).



# Connection Listing

## Chapter 4: Connection Listing

Introduction .....	125
Connection Listing Concepts .....	126
Definition of Terms .....	126
Logical vs. Physical Connection Listing .....	127
Connection Hierarchy .....	127
Special Reference Designators .....	129
Bus Notation .....	129
Properties .....	130
Implicitly Stepped Components .....	130
Installing the Connection Lister .....	131
Local Installation .....	131
SRM Installation .....	132
Installing Your Codeword .....	132
Using the Connection Lister .....	136
Using Associated Text .....	136
Archiving Drawings .....	137
Entering the Connection Lister Program .....	137
Running the Connection Lister .....	139
Connection Lister Default Files .....	142
Connection Lister Message File .....	146
Efficient Processing of Connection Lists .....	147
Connection Lister Output File .....	148
Connection Lister Operational Errors .....	156
Connection Lister Error File .....	157
Customizing the Connection Lister .....	159
Connection List Utilities .....	161
Merging Connection Lists .....	161
Generating a Rat's Nest .....	164
Comparing Connection Lists .....	169



# Connection Listing

Chapter

**4**

## Introduction

This chapter explains how to derive and use connectivity information from HP EGS drawings. Such connectivity information can be especially useful when comparing a printed circuit board drawing to its corresponding schematic drawing.

HP EGS provides a powerful family of connection listing tools that:

- Create connection lists from schematics and PC boards and note any connection errors
- Merge connection lists
- Generate a rat's nest (or "airline routing") of traces on a PC board
- Compare connection lists for logical equality

The first part of this chapter discusses the Connection Lister post-processor, which calculates connectivity from HP EGS artwork. You can then reformat the resulting connection list with a user-written program, and transfer it to another CAD system for simulation or analyses.

Later sections of the chapter explain how to use three HP EGS utilities to merge connection lists, generate a rat's nest, and compare connection lists.

## Connection Listing Concepts

### Definition of Terms

This chapter defines several terms:

Net	A list of ports connected by net elements.
Net Element	An HP EGS component used when calculating connectivity. Dimensions and associated components are not used. Ovals, arcs, and circles are approximated by polygons.
Net Name	Optional associated text with a tag value of 1002. The text string is associated to any net element and defines the net name for the component.
Instance	An HP EGS instance component in a drawing.
Part Name	An HP EGS library part's drawing name (e.g., dip8 or nand3). Each instance of a particular part has the same part name but a unique reference designator.
Reference Designator	Associated text with a tag value of 1001. The text string is associated to an instance, is required, and provides a unique name for each instance in the drawing (except unnamed wraps). A reference designator is always listed in a connection list. Instances without a reference designator will generate a warning message. The system recognizes three special reference designators: IGNORE, VIA, and SMASH (see the <i>Special Reference Designators</i> section of this chapter for more details).
Port	A named connection to a part; text is associated to an instance and has a tag value of 1003. The logical location of a port is a marker component (point) when calculating connectivity. The text string is the name of the port. A port can be defined on more than one layer by specifying logical layers. Multiple ports (logical and physical ports) can be defined at the same logical location, but the associated text must be added to different HP EGS layers.

## Logical vs. Physical Connection Listing

A connection is defined by either logical or physical rules. These rules are as follows.

### Logical Connection Rules

1. A connection exists if any of the following statements are true:
  - Components touch at at least one vertex
  - Two components cross and share a common vertex (i.e., a marker or a port)
  - Two components have the same net name
2. HP EGS layers are ignored; the system assumes all components exist in a common plane.
3. Text, note, and dimension components are not regarded as net elements.
4. Bus notation is allowed (when defining implicitly connecting components).

### Physical Connection Rules

1. A connection exists if either of the following statements is true:
  - Two components are on a common layer and touch in any manner
  - Two components are on different layers and touch the same port
2. Layer binding is used to implicitly copy components from the master layer to the bound layer.
3. Dimension components are disregarded.
4. Bus notation is not allowed.

## Connection Hierarchy

When a part is created in a drawing, it is either assigned a default logical level by HP EGS (as stated in a personality's process file) or explicitly assigned a logical level by you (using the LEVEL command). This logical level creates a hierarchy among the instances in a drawing. For example, instances at logical level 10 would be at a greater hierarchical level than instances at logical level 0. This can be useful if you want to look inside some instances for connections but not inside others. The hierarchy within a drawing can be used to control the operation of the Connection Lister, where you may set the smashing level and choose a flat vs. hierarchical listing.

### **Smashing Level**

Instances in a drawing can be smashed prior to calculating connectivity. This involves setting the smashing level option on the Connection Lister menu. As the Connection Lister processes a drawing, it automatically smashes all unnamed instances. It smashes named instances only if their logical level is less than or equal to the smashing level. (Note that this smashing level is sometimes referred to as the netting level.)

To ensure that you get unique net names and reference designators after smashing instances, the Connection Lister remembers the full reference designator path names for each remaining unsmashed instance. This path name is a concatenation of each reference designator encountered at each nesting level when smashing an instance. As usual, an instance without a reference designator will generate an error.

### **Flat vs. Hierarchical Listing**

The Connection Lister usually processes the contents of all instances as it looks for net elements; this process produces a “flat listing.” Optionally you can tell the Connection Lister to produce a “hierarchical listing” where the contents of some or all instances are not processed.

In a flat listing, reference designators found in nested instances are ignored. All net elements are assigned the same reference designator path name as the instance itself. To override this default behavior, you can set the Connection Lister’s menu option for Reference Designator Paths to Full Path Names; the result is path name concatenation similar to that occurring with smashed instances. The Full Path Names option also lets you use the three special HP EGS reference designators (IGNORE, VIA, and SMASH) on nested instances.

In a hierarchical listing, the Connection Lister generates an independent connection list for each instance in the drawing with a logical level greater than or equal to the hierarchical level. This hierarchy of lists considers only the external definition of each instance (i.e., ports and reference designators) and does not process the instance’s contents, as does flat listing. Note that because instances are also affected by the smashing level you set, an independent list is produced for each instance that satisfies the following condition:

$$\text{hierarchical level} \leq \text{instance logical level} \leq \text{smash level}$$



## Special Reference Designators

All instances in a drawing must have an assigned reference designator to avoid connection list errors. The following reference designators have special meaning for the Connection Lister:

- IGNORE** Instances with this reference designator are not used in any way when calculating connectivity. This feature lets you include library parts which are only used for aesthetic purposes in a drawing. The schematic parts GNDX and CHGNDX are two examples of this.
- SMASH** Instances with this reference designator are always smashed, regardless of the smashing level set at the time the drawing is processed. Thus, the system ignores the ports and properties of the instance and considers only the contents of the instance. You can assign this reference designator to an instance which is only used to combine other instances; the schematic library part GND is such an example.
- VIA** Instances with this reference designator are treated normally when generating connections, except the ports on the instances are not listed in the resulting connection list. This feature allows you to make physical interlayer connections without having each via appear in the connection list.

## Bus Notation

A bus is a net name that follows a special notation. The Connection Lister supports two types of bus notation: ranged and regular.

A ranged bus consists of nets that are named with consecutive numbers. For example, a net element with a net name of DATA[0:31] designates the ranged bus with the nets DATA[0] through DATA[31]. The lower bound of a ranged bus must always be less than or equal to its upper bound.

A regular bus designates a collection of net elements. Regular busses may be named or unnamed and may include ranged busses, taps off of ranged busses, and simple nets. For example, a net element with a net name of DATA[2:3],READ,WRITE designates an unnamed bus with nets named DATA[2], DATA[3], READ, and WRITE.

You can also name a regular bus so that it can be referred to as an entity. Use the = string to define a named bus. For example, a net element with a net name of CONTROL=READ,WRITE,DATA[0:1] defines a bus named CONTROL with the nets READ, WRITE, DATA[0], and DATA[1]. Use the ! string to declare the bus in another bus definition. For example, the bus CONTROL can be used in other bus definitions by specifying the bus name CONTROL!.

## Properties

A property is additional information you might want to include in a connection listing. Three property types are supported by the Connection Lister: Net, Instance, and Port. These properties are appended to the connection list output. You may put associated text for the properties on different layers to allow the selective output of the properties.

Properties are defined as follows:

- An instance property is defined by associating text with a tag value to the instance.
- A port property is defined by associating text with a tag value to the associated text defining the port name.
- A net property is defined by associating text with a tag value to a net element. A property is defined for each of the nets in a bus when the property is associated to the net element defining the bus.

## Implicitly Stepped Components

To assign a reference designator to each instance of an implicit step definition, you must associate text with a tag value of 1001 to the instance being stepped. The text string contains a list of reference designators rather than a single reference designator. Each reference designator in this list is assigned to an instance starting from the instance being stepped, and continuing with each row of instances in the step definition.

A list of reference designators can consist of ranged and simple names. For example, the reference designator text SHIFT[0:2], associated to an instance being implicitly stepped three times, designates the three instances as SHIFT0, SHIFT1, and SHIFT2. Likewise, a reference designator list of U1B,U2,SHIFTRREG[0:1] designates the four instances U1B, U2, SHIFTRREG0, and SHIFTRREG1. Letters can also be used as bounds for ranged names. For example, a reference designator list of U[A:C] designates the three instances UA, UB, and UC. The ordinal value of the lower bound of a ranged reference designator must always be less than or equal to the ordinal value of its upper bound.

Stepped components of the correct type may also be used as net elements. To assign a net name to a stepped net element, you may associate text with a tag value of 1002 to the net element being stepped. The text string contains a list of net names rather than a single name. Each name in this list is assigned to a net element starting from the net element being stepped, and continuing with each row of elements in the step definition. The rules for declaring unnamed busses apply for naming nets in a net element with a step definition. Because each net element in an implicit step definition may possess only one net name, busses may not be implicitly stepped.

In any implicitly stepped net element or instance definition, the stepped net elements and instances (“the children”) inherit all properties associated with the parent drawing.

## Installing the Connection Lister

The following sections describe how to install the Connection Lister post-processor on either a local disc or a Shared Resource Manager (SRM). Read only the section that pertains to your system. These instructions assume you have previously installed the HP EGS operating system and software.

If you have already installed the Connection Lister, skip to the section on *Using the Connection Lister* which follows this section.

### Local Installation

1. Make sure that the Graphics Editor and the General Drawing Personality are installed on your system in the volumes EWSYS: and EWCODE:. These should be installed before you can use the Connection Lister. To verify you have installed these items, refer to the verification procedure described in the *Installing HP EGS* manual.
2. Make sure that a volume named EWRC: exists on your local disc. This volume is where the Connection Lister software will reside. If no such volume exists, return to *Installing HP EGS* for instructions on creating a volume.
3. Insert the disc labeled EWRC1 in unit number 3. This is the right drive in an HP 9836A/C and the left drive in an HP 9121D or HP 82901A Flexible Disc drive.
4. Starting from the HP EGS Manager menu, use the arrow keys to select System Utilities, and press **Return**.

(If you are entering HP EGS through the Pascal 3.1 Operating System, simply proceed to step 6.)

5. Select the Pascal Command Line from this menu, and press **Return**.
6. Press **S** (for Stream) to initiate the installation procedure.
7. When the system prompts you for the name of the stream file, type in:

EWRC1:ewrc\_lin **Return**

Then follow the instructions displayed on the screen. The system will ask you to insert each disc and press the space bar to continue with the installation procedure. The system will copy a total of 3 discs into the volume EWRC:. When the procedure is finished, a message indicating successful completion will be displayed.

## SRM Installation

1. Make sure that the Graphics Editor and the General Drawing Personality are installed on your system in the directories EWSYS and EWCODE. These should be installed before you can use the Connection Lister. To verify that you have installed these items, refer to the verification procedure described in the *Installing HP EGS* manual.
2. Make sure that a directory named EWRC exists on your SRM. This directory is where the Connection Lister software will reside. If no such directory exists, return to *Installing HP EGS* for instructions on creating an SRM directory.
3. Insert the disc labeled EWRC1 in unit number 3. This is the right drive in an HP 9836A/C and the left drive in an HP 9121D or HP 82901A Flexible Disc drive.
4. Starting from the HP EGS Manager menu, use the arrow keys to select System Utilities, and press **Return**.

(If you are entering HP EGS through the Pascal 3.1 Operating System, simply proceed to step 6.)

5. Select the Pascal Command Line from this menu, and press **Return**.
6. Press **S** (for Stream) to initiate the installation procedure.
7. When the system prompts you for the name of the stream file, type in:

```
EWRC1:ewrc_sin Return
```

Then follow the instructions displayed on the screen. The system will ask you for the name of the system directory and the optional product directory (such as, #5:/HP\_EGS). Enter exactly the same information that was entered when you created the directory. The system will ask you to remove each flexible disc after everything is copied, insert the next disc, and press the space bar to continue with the installation procedure. The system will copy a total of 3 discs into the directory EWRC. When the procedure is finished, a message indicating successful completion will be displayed.

## Installing Your Codeword

The Connection Lister program requires a codeword in order to operate. Once you have installed the Connection Lister, you can follow the procedure in this section. If you live in the U.S. or Canada, the following procedure may require you to phone Hewlett-Packard during business hours (Mountain Time). If you live outside the U.S. or Canada, you must contact your HP Systems Engineer to obtain your codeword. It should take you about an hour to both obtain and install your codeword.

A codeword is a 16-digit number that you enter once to permanently link your Connection Lister software to a computer or HP-HIL ID Module. Codewords are also needed for the HP EGS personalities, the PHOTOPLOT/DRILL post-processor, and the IGES Translator. A unique codeword is needed for each workstation that operates the Connection Lister, including those working off of an SRM. A codeword is *not* a password that you type in each time you enter an HP EGS personality.

For the HP EGS 2.1 Connection Lister, your codeword will be the same one you installed for the HP EGS personalities. Locate the codeword you installed earlier, and install it again as instructed below for the Connection Lister. (If you have not yet obtained this HP EGS codeword, refer to the *Installing HP EGS* manual for further codeword installation instructions.)

---

#### Note

If you cannot locate the codeword you installed earlier, you can look at the codewords listed in the Graphics Editor's message file (EWCOD:gedit.ms.ASC) to find the correct number. Simply use the Editor to examine the contents of the file. Make no changes to this file at this time!

---

### Installing a Codeword from the HP EGS Manager Menu

There are two ways to enter the HP EGS codeword installation program: from the HP EGS Manager menu or from the Pascal Command Line. The instructions for the latter follow this section.

To install your HP EGS Connection Lister codeword from the HP EGS Manager menu, follow these steps. If you have any problems with the procedure, restart from step 1. To exit this utility, press **Stop**.

1. If your computer displays the HP EGS Manager menu:

```
==>System Utilities
    General Drawing
    Electrical Engineering
```

skip to step 2. If not, continue.

- a. Turn off your computer (your computer only; do not turn off the disc).
- b. Turn your computer back on. **Hold down the space bar.**
- c. When you see 1E SYSTEM\_EGS at the top right corner of your screen, lift the space bar and type in:

```
1E
```

The 1E will appear in the bottom right corner of your computer screen.

- d. The computer displays a screenful of HP EGS copyright information. Press the space bar to continue.
  - e. The computer prompts for the date. Enter the date and press **Return** or simply press **Return** to avoid the prompt.
  - f. The computer prompts for the time. Enter the time and press **Return** or simply press **Return** to avoid the prompt.
2. Select `System Utilities` from the HP EGS Manager Menu. To select a menu item, position the arrow by the menu item with the arrow keys. Press **Return**.
  3. Select `Install Codeword` from the System Utilities Menu. Press **Return**.

This utility runs a program to determine the product and serial numbers of your computer and/or any HP-HIL ID Modules. You will see a display similar to the one below.

```
CPU ID Prom - Product number : 9836C  Serial number : 2250A00749
```

The CPU ID Prom is the identifier of your computer. In the example above, the product number is 9836C and the serial number is the 10-digit number 2250A00749.

If you have more than one ID module or both a computer and an ID module, you must decide which of these you want to link to your copy of the Connection Lister. After you obtain your codeword you can use your copy of the Connection Lister only with the computer or HP-HIL ID Module you have chosen. The advantage of linking your copy of the Connection Lister to an ID module is that you can take your copy of the Connection Lister and your module to other workstations.

4. Write down the product and serial numbers of the computer or HP-HIL ID Module that you plan to link with your copy of the Connection Lister:

Product Number: \_ \_ \_ \_ \_ Serial Number: \_ \_ \_ \_ \_

5. If you live in the U.S. or Canada, phone the number shown in step 3c of your codeword certificate. Ask for the HP EGS Codeword Delivery Service.

If you live outside the U.S. or Canada, contact your HP Systems Engineer or HP Application Support Representative.

If you already have your codeword available (from previously installing the one needed to operate the HP EGS personalities), proceed to step 9.

6. The person you contact at Hewlett-Packard will ask for the:
  - Certificate serial number. This is listed at the top of your Codeword Certificate.
  - Product number of the device you are going to link to your copy of the Connection Lister. You wrote this down in step 5.
  - Serial number of the device you are going to link to your copy of the Connection Lister. You also wrote this down in step 5.

- The person will then give you your 16-digit codeword. Write it both below and on your certificate.

CODEWORD: \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

- The computer prompts you to enter a number between 1 and 4. These numbers represent the function for which you are installing the codeword. Enter:

2

to install the codeword for the Connection Lister, and press **Return**.

- Value in your codeword as prompted. Then press **Return**.
- The computer displays the name of the message file (`CON.MSG,ASC`) in which this utility will place your codeword. Simply press **Return**. Note: If you entered an SRM path name different than the one suggested in this manual, enter this path name in place of the one listed.

The system returns you to the System Utilities menu. To return to the HP EGS Manager menu, select `Return to Main Menu`.

The codeword should now be installed and you can use the Connection Lister software. Note that you must install one codeword for each computer on an SRM system.

### Installing a Codeword from the Pascal Command Line

If you are entering HP EGS through the Pascal 3.1 Operating System, follow these steps to install your codeword:

- Starting from the main Pascal command line, press **X** to execute a program.
- The system prompts for the name of the file you want to execute. Type in:

`EWCODE:codeword` for a local disc

`#5:/HP_EGS/EWCODE/codeword` for an SRM

If your directory path for an SRM differs from the one shown above, be sure to type in the correct path name for your system. Note that if you are installing a codeword on an SRM, no other users may be using HP EGS at that time.

- You have now entered the codeword installation program. From this point on, you may follow the instructions shown in the previous section, *Installing a Codeword from the HP EGS Manager Menu*, beginning with step 3.

## Using the Connection Lister

The Connection Lister works with the HP EGS Graphics Editor to help you derive connectivity information about a drawing and then correct any errors occurring within a drawing. A typical connection listing cycle involves:

1. Creating and saving a drawing (making a `_d` drawing file).
2. Archiving this drawing (making an `_a` archive file).
3. Running the Connection Lister program on this archive file (making `_c` and `_e` output and error files).
4. Re-entering the HP EGS Graphics Editor (through one of the personalities), and bringing up the original `_d` drawing file on the screen.
5. Inputting the Connection Lister's `_e` error file on the drawing.
6. Correcting any errors in the drawing (errors are described in the *Connection Lister Error File* section of this chapter).
7. Saving and archiving the drawing again, and re-running the Connection Lister program.

To help you better understand this cycle, this section describes in detail:

- Using associated text in HP EGS drawings
- Archiving drawings for connection listing
- Entering and running the Connection Lister program
- The Connection Lister default files and how to customize them
- The Connection Lister output and error files

### Using Associated Text

The Connection Lister looks at the associated text added to HP EGS drawings and components as it calculates connectivity within a drawing. This means that, when you make an HP EGS drawing, you should name instances, ports, and nets by adding associated text to them and a numeric tag which specifies the function of the text.

The currently defined tag field values used in connection listing are:

Value	Use	
1001	Reference designators	} Required tag values
1002	Bus/Net names	
1003	Port names	
1004	Part types	Optional tag value



Associated text and tag values are added to the drawing using the ADD TEXT/NOTE command. The following options are useful for this process:

- :AC Associates the text with a particular component in the drawing being edited.
- :AD Associates the text with the drawing being edited. This enables you to specify default values when building parts. When an instance of such a part is added to a drawing, all of its :AD text is copied to the drawing as :AC text, associated with the instance of the part.
- :D Causes the text to be displayed. If :D is not present, the text is added but not displayed. WINDOW :A can be used to display all “undisplayed” text.
- :L Specifies the logical location for the associated text and, optionally, the layers on which that logical location is defined.
- :T Specifies a tag value which defines the function of the associated text. The tag numbers 1000-1999 are reserved for use in HP-supplied library parts. We recommend that you use different numbers to define your own tag values.

For more information on how to place the associated text and other options, see the ADD NOTE/TEXT command in the *HP EGS Syntax Reference*.

Use the MOVE command to change the location of the associated text, and the MODIFY command to change other associated text characteristics. More information on the MOVE and MODIFY commands appears in the *HP EGS Syntax Reference*.

## Archiving Drawings

The HP EGS Connection Lister takes as input an archive file of an HP EGS drawing. Only files archived in HP EGS 2.0 or 2.1 may be processed in the Connection Lister. Be sure the process file used to create the drawing is loaded into HP EGS before archiving the drawing. If your drawings are stored as EGS/200 versions 1.0 or 1.05, just re-archive them in HP EGS 2.0 or 2.1. For EGS/45 archive files, you must retrieve and re-archive these drawings in HP EGS 2.0 or 2.1.

When archiving an HP EGS file for connection listing (with the ARCHIVE command), use the :P option which archives the process file along with the drawing itself, and the :M option which converts macro instances into ordinary instances. For further details on using the ARCHIVE command, refer to the *HP EGS Syntax Reference*.

## Entering the Connection Lister Program

You may run the Connection Lister program from either the HP EGS Electrical Engineering menu or from the Pascal Command Line. The following procedures explain both options.

### Operating from the HP EGS Electrical Engineering Menu

1. Select the System Utilities option from the HP EGS Manager menu.
2. Select the Filer from this menu.
3. Press **P** to set the system prefix.
4. Type the path name to the volume or directory where the Connection Lister program resides:

EWRC: for a local disc

#5:/HP\_EGS/EWRC for an SRM

If your directory path name for an SRM differs from the one shown above, be sure to type the correct path name for your system.

5. Press **Q** to leave the Filer.
6. Select the option named Return to Main Menu. The HP EGS Manager menu will appear.
7. Select the `Electrical Engineering` option from the HP EGS Manager menu.
8. Select either the `Electrical Schematic Connection Lister` or the `Printed Circuit Board Connection Lister`, depending on the type of connection list you want to create. The system will automatically load and execute the Connection Lister program you choose. It takes about 30 seconds to load the program.

The Connection Lister program is now ready to accept input from you.

### Operating from the Pascal Command Line

1. Select the System Utilities option from the HP EGS Manager menu.
2. Select the Filer from this menu.
3. Press **P** to set the system prefix.
4. Type the path name to the volume or directory where the Connection Lister program resides:

EWRC: for a local disc

#5:/HP\_EGS/ERWC for an SRM

If your directory path name for an SRM differs from the one shown above, be sure to type the correct path name for your system.

5. Press **Q** to leave the Filer. The System Utilities menu will appear.
6. Select the Pascal Command Line from this menu.
7. Press **X** to execute a program.

8. When the system asks for the name of the code file, type:

```
verify 
```

It takes about 30 seconds to load this code file.

The Connection Lister program is now ready to accept input from you.

### If You Get a Warning Message . . .

If you encounter a warning message at this point telling you the system cannot find either your codeword or your security device, check to make sure that:

- Your HP EGS Connection Lister codeword has been installed in the message file EWRC:con\_ms.ASC (for a local disc) or #5:/HP\_EGS/EWRC/con\_ms.ASC (for an SRM). Simply use the Editor to examine the contents of the file, and look for your 16-character codeword under the heading CODEWORDS. If no codeword exists there, return to the section entitled *Installing Your Codeword* in this chapter.
- Your external HP EGS ID Module (HP 46084A) is attached to your HP Series 300 computer. If not, attach the module to the computer.

## Running the Connection Lister

When you enter the Connection Lister program, the following happens:

1. The system asks for the name of the Connection Lister's message file:

```
Enter MESSAGE FILE name ?
```

If you have entered the program via the HP EGS Electrical Engineering menu, the system will provide a default answer for this question. However, if you have entered the program via the Pascal Command Line, respond by typing:

```
con
```

It is not necessary to type the `_ms.ASC` suffix for the file name, as the system will append this suffix for you if it is not present in your entry.

---

#### Note

If you are not prefixed to the Connection Lister's volume or directory (EWRC: or #5:/HP\_EGS/EWRC) before you enter the Connection Lister program, you must type complete path names (not just file names) in response to many Connection Lister prompts.

---

2. The system asks for the name of the file containing default values to be used when you run the Connection Lister.

Enter DEFAULT FILE name ?

If you have entered the program via the HP EGS Electrical Engineering menu, the system will provide a default file name for you, depending on the menu option you selected:

schphy for the Electrical Schematic Connection Lister

pcbphy for the Printed Circuit Board Connection Lister

Although a third default file (named `schlog`) exists for looking at logical (rather than physical) connections in a schematic, this file name will not appear automatically when you select the Electrical Schematic Connection Lister. If you want this default file instead of the one listed, you may edit the file name shown on the screen.

If you entered the program via the Pascal Command Line, you must type the name of the file you want to use. It is not necessary to type the `.ASC` suffix, as the system will append this suffix for you if it is not present in your entry.

If you have created a customized default file and want to use it instead of the ones available in HP EGS, simply backspace over the default entry and type the new file name.

3. The system asks which archived drawing file you want to connection list:

Enter ARCHIVE FILE name ?

Type the name of your archive file. It is not necessary to include the suffix `_a` in your entry. If you have placed the name of this file in a customized default file, the file name will automatically appear on the screen.

---

**Note**

The archive file you use should include the process file for the drawing and have smashed macro instances. These conditions are present in drawings archived with the `:P` and `:M` options of the ARCHIVE command.

---

4. The system asks where it should send the Connection Lister output:

Enter OUTPUT DESTINATION ?

If you press **Return** the Connection Lister output is placed in the same volume as the input file. If you want the output file in another volume, enter the volume name or directory path now.

5. The system then presents a menu listing the available Connection Lister options. To enter or change any values, simply use the arrow keys to select an item, and press **Return**. Another prompt will appear telling you what information the system needs from you.

The menu options are:

```

QUIT
Re-set to Default File      =
Physical or Logical Rules: =
Flat or Hierarchical List  =
Reference Designator Paths =
Smash to Logical Level     =
Components to Include      =
Net Properties Tag List     =
Port Properties Tag List   =
Instance Properties Tag List =
Net or Instance List Order =
Output Destination         =
Archive File to Process    =
RUN Connection Lister

```

The system displays the default values it finds in the specified default file on the menu. It also displays the names of the output volume and archive file you entered in steps 3 and 4. This gives you another chance to change these entries if desired.

The second line of the menu gives you the chance to change the name of the default file and/or re-set all the values displayed in the menu to those specified in the default file.

The HP EGS default files for the Connection Lister are described in the next section of this chapter. Refer to that section for a thorough discussion of each connection listing option.

6. Once you are satisfied with the items listed in the default menu, select the option named

```
RUN
```

to generate the connection list. The program will output dots on the screen to indicate that it is processing the archive file. If during this process you want to stop the program, simply press **Stop** to return to the Connection Lister menu.

7. To exit the Connection Lister, select the option named

```
QUIT
```

to return to the HP EGS Electrical Engineering menu.

---

**Note**

If you decide to re-run the Connection Lister program on a given archive file (perhaps to generate both a net-oriented and an instance-oriented connection listing from the same drawing), the program will read the entire archive file over again. It does not keep a copy of the archive file in current memory.

Re-running the program will also overwrite the original connection list and error files, unless the output destination is changed.

---

## Connection Lister Default Files

A default file is a message file that contains specific information telling the system how to create a connection list. The three default files supplied with the HP EGS Connection Lister are:

EWRC:schphy.ASC	for the Electrical Schematic Connection Lister
EWRC:schlog.ASC	for the Electrical Schematic Connection Lister
EWRC:pcbphy.ASC	for the Printed Circuit Board Connection Lister

The Electrical Schematic default file for creating a physical connection list (EWRC:schphy.ASC) contains the following values:

```

phys_log_def      ^D^2
net_inst_def      ^D^1
flat_hier_def     ^D^2
ref_desig_def     ^D^1
cmpnt_filter_value ^D^#E -E7 -E51
hier_def_value    ^D^32767
smash_value       ^D^0
net_tags_value    ^D^
port_tags_value   ^D^
inst_tags_value   ^D^1004
output_dest_value ^D^CONSOLE:
archive_file_value ^D^

```

When you run the Connection Lister using this message file, the following values will be displayed on your screen:

```

Re-set to Default File      = EWRC:schphy.ASC
Physical or Logical Rules: = Logical
Flat or Hierarchical List  = Hierarchical, 32767
Reference Designator Paths = External only
Smash to Logical Level     = 0
Components to Include      = #E -E7 -E51
Net Properties Tag List    =
Port Properties Tag List   =
Instance Properties Tag List = 1004
Net or Instance List Order = Net
Output Destination        = CONSOLE:
Archive File to Process    =

```

The Electrical Schematic default file for creating a logical connection list (EWRC:schlog.ASC) contains the following values:

```

phys_log_def      ^D^2
net_inst_def      ^D^1
flat_hier_def     ^D^2
ref_desig_def     ^D^1
hier_def_value    ^D^32767
smash_value       ^D^0
cmPnt_filter_value ^D^#E -E7 -E50
net_tags_value    ^D^
port_tags_value   ^D^
inst_tags_value   ^D^1004
output_dest_value ^D^CONSOLE:
archive_file_value ^D^

```

When you run the Connection Lister using this message file, the following values will be displayed on your screen:

```

Re-set to Default File      = EWRC:schlog.ASC
Physical or Logical Rules: = Logical
Flat or Hierarchical List  = Hierarchical, 32767
Reference Designator Paths = External only
Smash to Logical Level     = 0
Components to Include      = #E -E7 -E50
Net Properties Tag List    =
Port Properties Tag List   =
Instance Properties Tag List = 1004
Net or Instance List Order = Net
Output Destination        = CONSOLE:
Archive File to Process    =

```

The Printed Circuit Board default file for creating a physical connection list (EWRC:pcbphy.ASC) contains the following values:

```

net_inst_def          ^D^1
flat_hier_def         ^D^1
phys_log_def          ^D^1
ref_desig_def         ^D^1
hier_def_value        ^D^0
smash_value           ^D^0
cmpnt_filter_value    ^D^-E #I #E51 #E52 #E53 #E54 #E5 #E6 #E7 #E8
                      #E11 #E12 #E13 #E14
net_tags_value        ^D^
port_tags_value        ^D^
inst_tags_value        ^D^1004
output_dest_value     ^D^CONSOLE:
archive_file_value     ^D^

```

When you run the Connection Lister using this message file, the following values will be displayed on your screen:

```

Re-set to Default File      = EWRC:pcbphy.ASC
Physical or Logical Rules:  = Physical
Flat or Hierarchical List   = Flat
Reference Designator Paths  = Full Path
Smash to Logical Level      = 0
Components to Include       = -E #I #E51 #E52 #E53 #E54 #E5 #E6 #E7 #E8
                          #E11 #E12 #E13 #E14
Net Properties Tag List     =
Port Properties Tag List    =
Instance Properties Tag List = 1004
Net or Instance List Order  = Net
Output Destination          = CONSOLE:
Archive File to Process     =

```

You can change any of the information in these files to customize them for your application.

These files are message files, where each entry follows this format:

```
tag ^D^ information
```

You may change the messages in the information field, but you should not change the message tags or the delimiter (^D^). For more information on message files, please refer to Chapter 2, *Customizing Your System*, in this manual.



The following table explains the meaning of each tag in a Connection Lister default file and its acceptable range of values.

phys_log_def	Specifies to use logical or physical connection list rules. A value of 1 indicates physical, and 2 indicates logical.
flat_hier_def	Specifies to make either a flat or hierarchical connection list. A value of 1 indicates flat, and 2 indicates hierarchical.
hier_def_value	Specifies the level at which you want the system to stop netting individual parts in the drawing. This number is an integer. Any part with this specified logical level or with a higher logical level is independently netted to its own logical level. Parts which only have instances with IGNORE or VIA reference designators are not netted. With a hierarchical connection list, the system does not look for net elements inside instances at the netting level or at a lower logical level.
ref_desig_def	Specifies whether or not you want the system to consider the reference designators found on subinstances in the process of smashing instances to look for net elements in a flat connection list. A value of 1 indicates external values only, and 2 indicates full path names.
smash_value	Indicates the logical level to which the system will smash instances before it looks for parts and net elements. Any part in the connection list will thus have a logical level equal to or less than the one specified in this option. The system stops smashing an instance when its logical level is equal to or less than this logical level. However, the system always smashes unnamed instances when it encounters them. The smashing process does not affect the input drawing as the system only logically performs this operation. The system accepts values between 0 and $\pm 32767$ .
cmpnt_filter_value	Specifies the components and layers which may be processed. Placing + and # in front of an HP EGS component descriptor indicates that you want to include all of those components; - indicates that you want to include none of those components. Placing :B after a component descriptor and layer number indicates that you want to include any layers bound to this descriptor and layer number.
net_tags_value	Specifies the associated text tag values to include as net properties in the connection list output. These properties are listed in the order in which their tag values appear in the tag list. If a property is not found for a tag value in the list, an empty property field is output. Properties with the same tag value are ordered by the layer numbers of their text. The system accepts tag values between 0 and $\pm 32767$ .

- `port_tags_value` Specifies the associated text tag values to include as port properties in the connection list output. These properties are listed in the order in which their tag values appear in the tag list. If a property is not found for a tag value in the list, an empty property field is output. Properties with the same tag value are ordered by the layer numbers of their text. The system accepts tag values between 0 and  $\pm 32767$ . Note: A tag value of  $-32768$  is a special case. It will output the approximate number of bytes of memory that was required by the system to generate the connection list.
- `inst_tags_value` Specifies the associated text tag values to include as instance properties in the connection list output. These properties are listed in the order in which their tag values appear in the tag list. If a property is not found for a tag value in the list, an empty property field is output. Properties with the same tag value are ordered by the layer numbers of their text. The system accepts tag values between 0 and  $\pm 32767$ .
- `net_inst_def` Specifies to make either a net- or instance-oriented connection list output file. A value of 1 indicates a net-oriented output file; 2 indicates an instance-oriented output file.
- `output_dest_value` The complete volume specification where you want to send the connection list output and its error file.
- `archive_file_value` The complete file specification of the archived drawing file to process.

Note that part names, reference designators, net and bus names, and port names always appear in a connection list.

## Connection Lister Message File

The Connection Lister message file (EWRC:con\_ms.ASC) contains messages needed for program operation. This file must always be on-line when you run the Connection Lister.

There are several entries in this file you may want to change. These include the codewords and names of stroke tables referenced by your drawings.

### Codewords

Security codewords are needed to execute the Connection Lister program. You must install a codeword before you can run the program. At times it is convenient to enter codewords in this file using the Editor rather than the installation program described at the beginning of this chapter. This is particularly true if you are installing many codewords at once, as may be the case for SRM users sharing the system.

If you edit the codewords in this file, be sure you simply exchange items character-for-character within the CODEWORD section of the file. Adding more characters will require that you run the Message File Utility available from the HP EGS System Utilities menu. If you need to do this, refer to the section explaining editing and indexing a message file in Chapter 2, *Customizing Your System*, in this manual.

### Stroke Tables

The Connection Lister program needs to know which stroke tables for notes and text to use when it generates a connection list.

This information is important when running a physical connection list, as notes and text can touch other components in a drawing. Default stroke tables are listed for you in the Connection Lister's message file. These are the standard HP EGS default tables EWSYS:note\_st.TEXT and EWSYS:text\_st.TEXT. It is important that the stroke tables listed in the message file match those used when a drawing was created. If they do not match, the program could calculate incorrect connectivity for the drawing. This also means that the path names to these files must be correct for your system, and that these stroke files must be on-line when you run the Connection Lister.

If you have used many different stroke tables to create drawings, you may want to copy and customize this message file for your applications. Remember that if you make any changes to this file, you may need to run the Message File Utility available from the HP EGS System Utilities menu. For more information, please refer to the section explaining editing and indexing a message file in Chapter 2, *Customizing Your System*, in this manual.

### Efficient Processing of Connection Lists

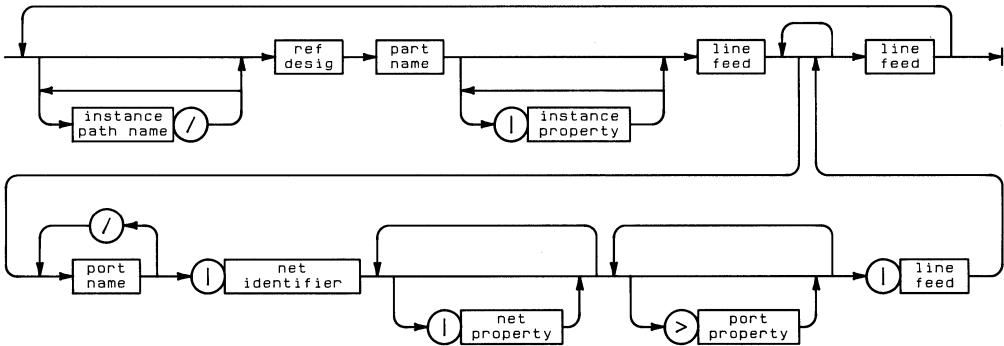
Here are some hints for the efficient processing of connection lists:

- Keep the drawing size within the range of  $\pm 32767$  system units.
- Avoid using circles, arcs, and ovals as net elements.
- Keep as many net elements as possible at the first nesting level of your drawing.
- Avoid segmenting connection lines. Use one complete line rather than many shorter lines with common endpoints.
- Avoid using unnecessary markers. Remember that a marker is needed only to connect crossing components when netting with logical rules.
- Use master layers to specify connections between two layers.
- Specify a hierarchical connection listing when instances in the drawing to be netted do not contain net elements.
- If you are not using notes and text as connective elements, you can either remove the stroke table references from the Connection Lister's message file (EWRC:con\_ms.ASC), or not have these files on-line.

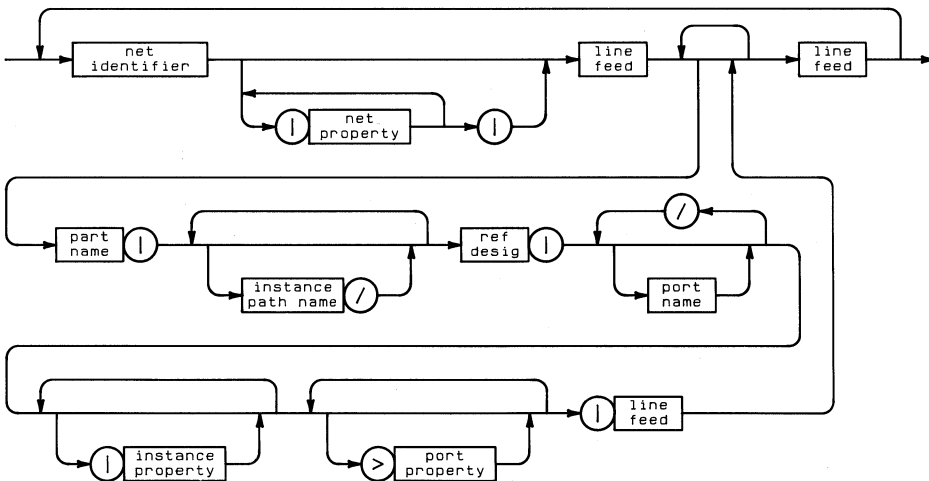
## Connection Lister Output File

The output of the Connection Lister is a text file that can be edited or reformatted for further processing by a user program. The name of a connection list output file consists of the name of the drawing being netted with a suffix `_.c`. This output file contains a connection list showing the connectivity of the drawing in terms of nets, instances, and ports.

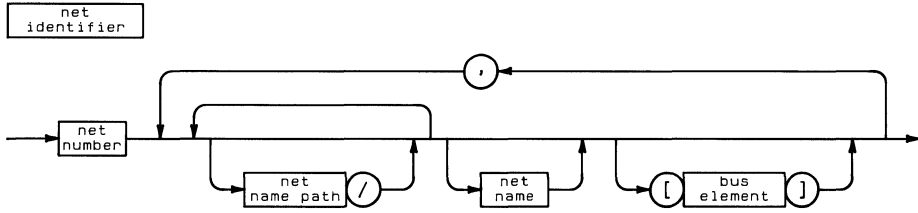
This connectivity can be described in either a net-oriented form or an instance-oriented form. A net-oriented description lists all nets and, for each net, the instances to which it connects. An instance-oriented description lists all instances and, for each instance, the nets to which it connects. In either case each net is assigned a unique net number based on the order in which the nets are processed. This ordering varies depending on the exact layout of the drawing data in the archive file. The net and instance connectivity descriptions take the following form:



**Instance-oriented Connection List Format**



**Net-oriented Connection List Format**



Item	Description/Default	Range Restrictions
net property	Associated text string listed for each tag value shown on the menu.	250 characters maximum
line feed	Start of a new line.	
part name	Name of the instance's drawing file excluding the suffix.	8 characters maximum
instance path name	Concatenated reference designators encountered at each nesting level when smashing an instance; appears if the smashing level option is set to Full Path Names.	250 characters maximum
ref desig	Reference designator; associated text required for each instance.	Must have the tag value 1001; 250 characters maximum
port name	Associated text defining a port on an instance.	Must have the tag value 1003; 250 characters maximum
instance property	Associated text string listed for each tag value shown on the menu. HP EGS library parts use tag value 1004.	250 characters maximum
port property	Associated text string listed for each tag value shown on the menu.	250 characters maximum
net number	Integer assigned by the system.	1 through the number of nets in the drawing
net name path	Concatenated reference designators encountered at each nesting level when smashing an instance; appears if the smashing level option is set to Full Path Names.	
net name	Optional user-supplied associated text for naming nets.	Must have the tag value 1002; 250 characters maximum
bus element	Identifier for one net of a bus, like DATA [3].	Bus naming conventions

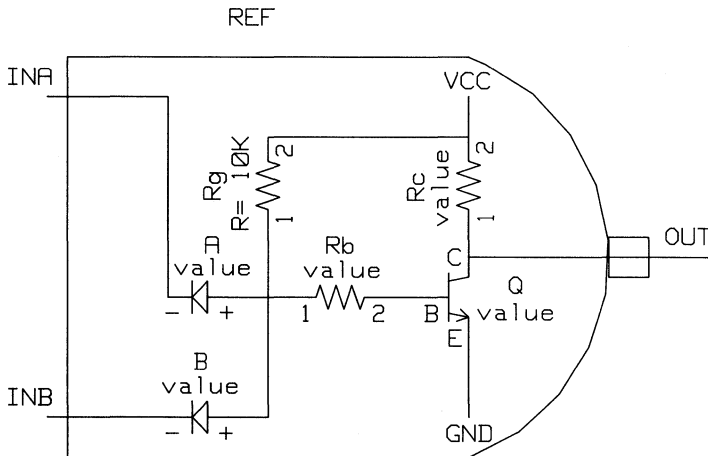
## 150 Connection Listing

Flat and hierarchical connection lists can be in either an instance- or a net-oriented form. The net or instance connectivity descriptions in a connection list of a part are alphabetically ordered by the net names or the instance reference designators. Within the connection list of a part, the system automatically assigns a number to each of the nets. In a hierarchical connection list, the connection lists are first ordered by the logical level of each part, then by the nesting level of each part within the drawing.

The format of a flat connection list file is:

```
header that includes: {drawing_name logical_level
errors detected before extracting nets
net or instance connectivity description
errors detected after extracting nets
```

Careful examination of the next few examples will illustrate the differences. Let's look at the following drawing `exnand2_d` and two possible connection lists we could derive from it:



Example 1: This connection list is the result of processing `exnand2` with the following values set in the Connection Lister menu:

```

Re-set to Default File      = EWRC:schphy.ASC
Physical or Logical Rules  = Logical
Flat or Hierarchical List  = Flat
Reference Designator Paths = External only
Smash to Logical Level     = 0
Components to Include      = #E -E1 -E7 -E51
Net Properties Tag List    =
Port Properties Tag List   =
Instance Properties Tag List = 1004
Net or Instance List Order = Net

```

This menu differs from the default file's values in only two ways: the resulting list will be flat, and components on layer 1 will be excluded.

```

!exnand2 1

1
  resis!Rb!2!value!
  n!Q!B!value!

2
  diode!B!-!value!

3
  diode!A!-!value!

4
  n!Q!C!value!
  resis!Rc!1!value!

5
  diode!A!+!value!
  resis!Rb!1!value!

6
  diode!B!+!value!
  resis!Rg!1!R= 10K!

7 GND
  n!Q!E!value!

8 VCC
  resis!Rc!2!value!
  resis!Rg!2!R= 10K!

```

## 152 Connection Listing

Example 2: This example differs from Example 1 in only one way: the list is instance-oriented rather than net-oriented.

```
lexnand2 1

A diode!value!
  -!2!
  +!6!

B diode!value!
  -!5!
  +!7!

Q nPn!value!
  C!1!
  E!4 GND!
  B!8!

Rb resis!value!
  1!6!
  2!8!

Rc resis!value!
  1!1!
  2!3 VCC!

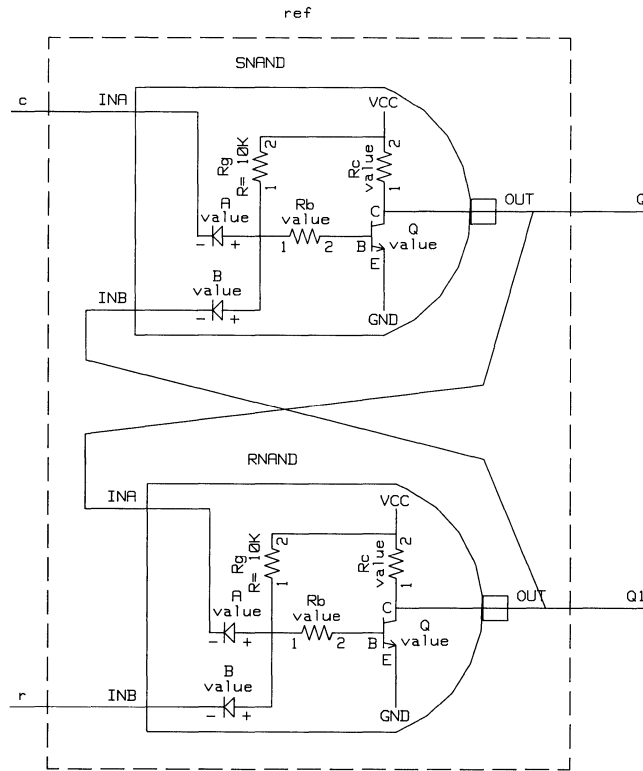
Rd resis!R= 10K!
  2!3 VCC!
  1!7!
```

The format of a hierarchical connection list file is similar to that of a flat connection list. The difference is that a hierarchical connection list is a collection of flat list files for each instance that satisfies the condition:

hierarchical level  $\leq$  instance level  $\leq$  smash level



Now let's look at the drawing `exsrff_d` and two possible connection lists we could derive from it:



Example 3: This example differs from Example 1 in two ways: the list is hierarchical rather than flat, and the hierarchical level is set to 0.

```

|diode 0
|nPN 0
|resis 0
|exsrff 2
1
resis!RNAND/Rb!2!value!
nPN!RNAND/Q!B!value!

```

## 154 Connection Listing

2 diode:SNAND/A|-!value!  
3 diode:RNAND/A|+!value!  
resis:RNAND/Rb!1!value!  
4 diode:RNAND/B|+!value!  
resis:RNAND/Rg!1!R= 10K!  
5 diode:SNAND/B|+!value!  
resis:SNAND/Rg!1!R= 10K!  
6 nen:RNAND/Q!C!value!  
resis:RNAND/Rc!1!value!  
diode:SNAND/B|-!value!  
7 diode:RNAND/B|-!value!  
8 nen:SNAND/Q!C!value!  
resis:SNAND/Rc!1!value!  
diode:RNAND/A|-!value!  
9 diode:SNAND/A|+!value!  
resis:SNAND/Rb!1!value!  
10 resis:SNAND/Rb!2!value!  
nen:SNAND/Q!B!value!  
11 RNAND/GND  
nen:RNAND/Q!E!value!  
12 RNAND/VCC  
resis:RNAND/Rc!2!value!  
resis:RNAND/Rg!2!R= 10K!  
13 SNAND/GND  
nen:SNAND/Q!E!value!  
14 SNAND/VCC  
resis:SNAND/Rc!2!value!  
resis:SNAND/Rg!2!R= 10K!

Example 4: This example differs from Example 3 in only one way: instances are smashed to logical level 2 rather than logical level 0.

```

|resis 0

|diode 0

|nPN 0

|exnand2 1

1
    resis|Rb|2|value|
    nPN|Q|B|value|

2
    diode|B|-|value|

3
    diode|A|-|value|

4
    nPN|Q|C|value|
    resis|Rc|1|value|

5
    diode|A|+|value|
    resis|Rb|1|value|

6
    diode|B|+|value|
    resis|Rd|1|R= 10K|

7 GND
    nPN|Q|E|value|

8 VCC
    resis|Rc|2|value|
    resis|Rd|2|R= 10K|

|exsrf 2

1
    exnand2|SNAND|INA||

2
    exnand2|RNAND|OUT||
    exnand2|SNAND|INB||

```

```
3      exnand2!RNAND!INB!!
```

```
4      exnand2!SNAND!OUT!!  
      exnand2!RNAND!INA!!
```

### Connection Lister Operational Errors

The following errors will terminate processing and return you to the Connection Lister menu, after giving you a warning message. The first line of each entry shows the literal message you will see on the screen, while the paragraphs following it further explain what the message means.

- CONERR1      Error while reading from the message file.

An error occurs while the system is reading the CONCNST constants from the Connection Lister's message file. You should verify that all the Connection Lister constants and error messages are present in the message file. The program terminates.
- CONERR2      The file (file name) could not be created.

The system cannot create the connection list output file or an error file. The program terminates.
- CONERR3      The length of the reference designator "(reference designator)" is too long.

The maximum length of a reference designator without its reference designator path name is 250 characters. The reference designator is truncated when this length is exceeded. This error can occur when the length of the missing reference designator string is too long.
- CONERR4      Out of memory.

Not enough memory is available for the system to complete the connection list. The program terminates.
- CONERR5      Unexpected error.

An unexpected error is encountered in the Connection Lister. Please report the error trace information outputted by the system. The program terminates.

## Connection Lister Error File

An error file is an input file for the HP EGS Graphics Editor. It consists of notes which identify the location and type of error encountered when calculating connectivity. The name of the error output file includes the name of the part being netted and the suffix `_e`, and is created on the same volume as the connection list output file. For most of the errors you might encounter, the system is still able to generate the connection list.

An error output file contains the ADD NOTE commands for layer 255 (the error message layer). If you enter the HP EGS Graphics Editor and input this error file on your drawing, the notes will show you the location and tell you the nature of each error. In hierarchical connection lists, the system may generate an error output file for each part netted (one error file per original drawing file). The locations of the notes in each error output file are based on the units of each drawing.

The notes that describe connectivity errors are as follows. The first line of each entry shows the literal message you will see on the screen, while the paragraphs following it further explain what the message means.

- CONERR6 Duplicate reference designator. Creating (reference designator).  
The reference designator of an instance may be listed with a DRD suffix. This suffix means that the same reference designator was found on another instance. These suffixes have an integer attached to them (like DRD20). Note that this error does not apply to instances with IGNORE, SMASH, or VIA reference designators.
- CONERR7 Missing reference designator. Creating (reference designator).  
An instance with no reference designator is assigned a reference designator of MRD. These reference designators have an integer attached to them (like MRD20).
- CONERR8 Invalid reference designator in list (reference designator list).  
A reference designator with an invalid ranged name is found in a list of reference designators associated to an implicitly stepped instance. The syntax of the ranged name and its bounds should be checked. The invalid reference designator is ignored.
- CONERR9 Invalid bus or bus element declaration: (bus or bus element declaration).  
An invalid ranged bus or bus element declaration is found. The syntax of the ranged name and its bounds should be checked. The invalid bus or bus element is ignored.

- CONERR10 Recursive bus definition: (first named bus), (second named bus).  
A named bus is defined recursively. The recursive definition in the bus declaration is ignored.
- CONERR11 Bus declaration not allowed with physical connection rules: (bus declaration).  
A bus declaration is found in the drawing being netted with physical rules. The bus declaration is ignored.
- CONERR12 Bus declaration not allowed in implicitly stepped net elements: (bus declaration).  
An implicitly stepped net element contains a named bus declaration. The named bus declaration is ignored.
- CONERR13 Net has been named multiple times with different names.  
A net may have multiple different names only when the names have different reference designator path names. An error is reported when a net is found with two different names with the same reference designator path names. This error can be detected when a net possesses multiple names with the same path name in a connection list.
- CONERR14 Invalid bus tap.  
A net or a bus is tapped off of a bus whose declaration does not include the desired net names.
- CONERR15 Net is not connected to any ports.  
None of the net elements of a net is connected to a port.
- CONERR16 Part has no ports.  
A part has no ports defined. The part definition is ignored.
- CONERR17 Too many net names in list: (net name list).  
An implicitly stepped net element has too many net names for the number of elements in the step specification. The extraneous net names are ignored. If the number of net names is less than the number of elements, some net elements will be unnamed.  
This error is also output when more than one net name is associated to a net element. The extraneous net name definitions are ignored.

- CONERR18** Too many reference designators in list: (reference designator list).  
 An implicitly stepped instance has too many reference designators for the number of instances in the implicit step specification. The extraneous reference designators are ignored.  
 Some instances will have an MRD reference designator if the number of reference designators in a reference designator list is less than the number of implicitly stepped instances.  
 This error is also output when more than one reference designator is associated to an instance. The extraneous reference designator definitions are ignored.
- CONERR19** Named bus definition not found: (named bus).  
 The definition of a named bus that is used in another bus declaration is not found. The named bus is ignored.
- CONERR20** Port is not defined on any layer: (port name).  
 The specified port is not defined on any layer. The logical layers of a port are defined with the :L option of the ADD TEXT command. This error is generated only when netting with physical rules. The port definition is ignored.
- CONERR21** Duplicate named bus definition: (named bus).  
 A named bus has more than one definition. The last bus definition will be used.
- CONERR22** Part has unconnected ports.  
 A part has one or more ports which are not connected to any net. This is more a warning than an error. The message may be suppressed by attaching any net element (perhaps a marker) to the offending port. This message appears at the end of the connection list file. Post-processors such as `Compare Connection Lists`, `Merge Connection Lists`, and `Generate Rat's Nest` ignore this message.

## Customizing the Connection Lister

Two things you may want to customize for your own application are the screen messages that appear during operation and the default string constants used by the program. The following sections describe how to change these items.

### Screen Messages

The Connection Lister message file (EWRC:con\_ms.ASC) contains most of the menu information and screen messages that a user sees while running the program. You may want to customize this information for foreign language applications, or to rewrite the messages that appear. Remember that if you make any changes to this file, you may need to run the Message File Utility available from the HP EGS System Utilities menu. For more information, please refer to the section explaining editing and indexing a message file in Chapter 2, *Customizing Your System*, in this manual.

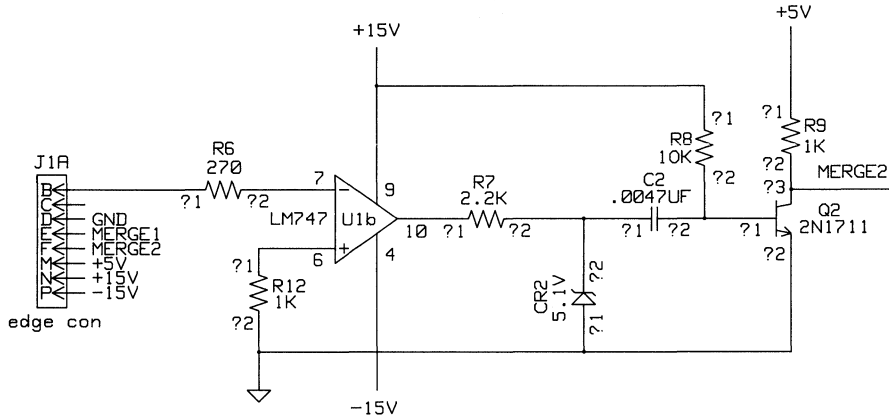
### Default String Constants

The system uses string constants for various purposes. If you wish to change any of the defaults, you must edit the Connection Lister's message file (EWRC:con\_ms.ASC) and modify the messages with the labels specified here:

CONCNST1	Default field separator.
CONCNST2 /	Default port name field separator.
CONCNST3 IGNORE	Default reference designator for instances that are to be ignored.
CONCNST4 DRD	Default duplicate reference designator error suffix.
CONCNST5 MRD	Default missing reference designator error suffix.
CONCNST6 >	Default port property name field separator.
CONCNST7 ,	Default net name field separator.
CONCNST8 /	Default path name field separator for reference designators and net names.
CONCNST9 SMASH	Default reference designator for instances that are to be logically smashed.
CONCNST10 =	Default named bus definition string.
CONCNST11 !	Default named bus use string.
CONCNST12 ,	Default net name and bus net name list separator.
CONCNST13 ,	Default reference designator list field separator.
CONCNST14 ERROR	Default error message.
CONCNST15	Default part name identifier.
CONCNST16 VIA	Default reference designator for instances which are not to appear in a connection list.







You could merge the two connection lists by:

1. Selecting the Electrical Engineering option from the main Manager menu, and then selecting the utility named Merge Connection Lists from the next menu.
2. Answering the prompts that appear on the screen as follows:

```

Enter the name of the message file: EWRC:outils
Enter the name of the output (merged) file: EXAMPLE:merge
Enter the name of the first connect list file: EXAMPLE:connex1
Enter the name of the next connect list file: EXAMPLE:connex2
Enter the name of the next connect list file:
Done.
    
```

The first prompt is answered by the system. The next three prompts are specified by you as appropriate. Note that file name suffixes need not be typed in, and that you must type in the complete file specification. Then press **Return** to indicate that there are no more entries. The system tells you when the connection list merger is done, and you are free to perform other HP EGS operations.

The merged connection file EXAMPLE:merge.c looks like this:

```

1
  opamp1:U1a:12:LM747:
  resis:R2:1:2.2K:

2
  edge1:J1B:B:edge con:

3
  vertres:R3:10K:
  cap:C1:1:0.0047UF:
  npn:Q1:1:2N1711:
    
```

4  
 resis:R1:2:270:  
 opamp:U1a:1:LM747:

5  
 vertres:R5:1:270K:  
 opamp:U1a:2:LM747:

6  
 zdiode:CR1:2:5.1V:  
 resis:R2:2:2K:  
 cap:C1:1:.0047UF:

7  
 edgse1:J1A:C:edge con:

8  
 opamp:U1b:10:LM747:  
 resis:R7:1:2.2K:

9  
 vertres:R12:1:1K:  
 opamp:U1b:6:LM747:

10  
 resis:R6:1:270:  
 edgse1:J1A:B:edge con:

11  
 zdiode:CR2:2:5.1V:  
 resis:R7:2:2K:  
 cap:C2:1:.0047UF:

12  
 resis:R6:2:270:  
 opamp:U1b:7:LM747:

13  
 vertres:R8:2:10K:  
 cap:C2:2:.0047UF:  
 nen:Q2:1:2N1711:

14  
 +15V  
 vertres:R3:1:10K:  
 cap:C7:2:.001UF:  
 opamp:U1a:13:LM747:  
 edgse1:J1B:N:edge con:  
 vertres:R8:1:10K:  
 opamp:U1b:9:LM747:  
 edgse1:J1A:N:edge con:

## 164 Connection Listing

```
15  +5V
    vertres|R4|?1|1K|
    edgse1|J1B|M|edgse con|
    vertres|R9|?1|1K|
    edgse1|J1A|M|edgse con|

16  -15V
    oPampF1|U1a|4|LM747|
    edgse1|J1B|P|edgse con|
    oPampF1|U1b|4|LM747|
    edgse1|J1A|P|edgse con|

17  GND
    zdiode|CR1|?1|5.1V|
    nPn|Q1|?2|2N1711|
    vertres|R5|?2|270K|
    cap|C7|?1|.001UF|
    edgse1|J1B|D|edgse con|
    zdiodeR2|?1|5.1V|
    nPn|Q2|?2|2N1711|
    vertres|R12|?2|1K|
    edgse1|J1A|D|edgse con|

18  MERGE1
    vertres|R4|?2|1K|
    nPn|Q1|?3|2N1711|
    edgse1|J1B|E|edgse con|
    edgse1|J1A|E|edgse con|

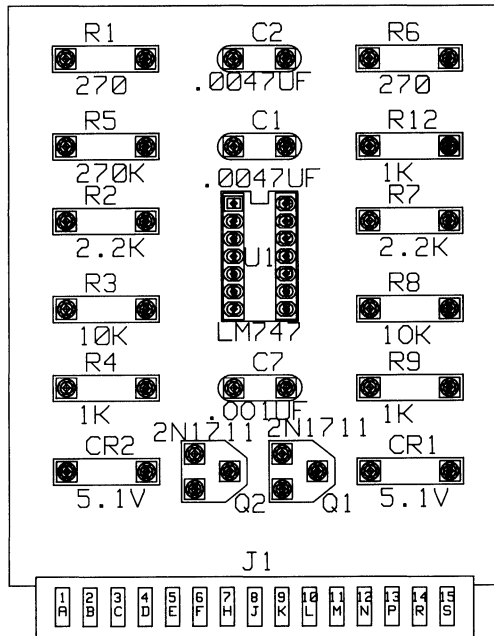
19  MERGE2
    edgse1|J1B|F|edgse con|
    vertres|R9|?2|1K|
    nPn|Q2|?3|2N1711|
    edgse1|J1A|F|edgse con|

20  REFERENCE SIGNAL
    resis|R1|?1|270|
    edgse1|J1B|C|edgse con|
```

### Generating a Rat's Nest

The Generate Rat's Nest program combines a connection list from a back-annotated schematic and an archive file from a placed PC board. It creates a text file full of ADD LINE commands that, when input, places lines connecting the proper points/ports on the PC board. You can map net numbers, net names, and/or net number ranges to different layers and line widths on the PC board.

Let's look at an example. Assume that you have placed and annotated all the physical parts required by the two schematic drawings referenced earlier (`connex1_d` and `connex2_d`) in a PC board drawing, as shown here:



You then can add the traces connecting the appropriate parts by generating a rat's nest or "airline routing" of the traces. To do so, you must:

1. Save the PC board drawing in a drawing file, and then archive the drawing using the `ARCHIVE :D1 :M` command (which ensures that instances of parts with logical levels less than or equal to 0 are not archived, and that macro instances are smashed).
2. Make sure the reference designators and physical port names used in the schematic drawing match those on the parts used in the PC board (for example, `U1A = U1, ?1 <> 1`). If not, back-annotate the schematic and rerun its associated connection list. If more than one schematic was used to create the PC board, merge all the connection lists.
3. Select the Electrical Engineering option from the main Manager menu, and then select the utility named `Generate Rat's Nest` from the next menu.

## 4. Answer the prompts that appear on the screen as follows:

```

Enter the name of the message file: EWRC:rnest
Enter the name of the schematic connect list file: EXAMPLE:merge
Enter the name of the PC board archive file: EXAMPLE:pcex
Enter the name of the output (rat's nest) file: EXAMPLE:pcrat

Enter a net number, a net number range, or a net name to be processed:
(ex: 5, 23-78, GND, *specifies 'the rest of the nets'. Enter an
empty line when done.) 1-11
Enter the layer number for the traces: (1 <= n <= 255) 5
Enter the width for the traces: (n >= 0) 15

Enter a net number, a net number range, or a net name to be processed:
(ex: 5, 23-78, GND, *specifies 'the rest of the nets'. Enter an
empty line when done.) *
Enter the layer number for the traces: (1 <= n <= 255) 6
Enter the width for the traces: (n >= 0) 15

Enter a net number, a net number range, or a net name to be processed:
(ex: 5, 23-78, GND, *specifies 'the rest of the nets'. Enter an
empty line when done.)

```

Note that a default answer appears for the first prompt. You need not type in file name suffixes, but you must type in complete file specifications.

## 5. Following the last entry, the system processes the information and gives these messages:

```

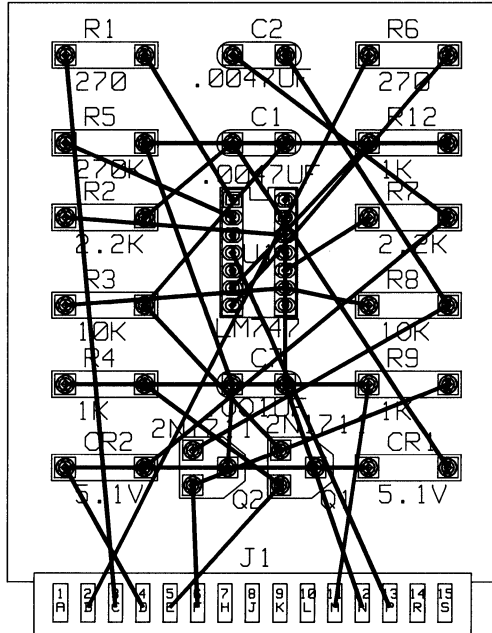
Processing the Schematic Connect List . . .
Processing the PC Board Archive File . . .
    Found $FILES list
Found EDIT pcex
Looking for ADDs of Library Parts . . . . .
Creating Output File . . .
Done.

```

If appropriate, the system also lists on the screen any associated text it was unable to find on the PC board. The final message indicates that the system has stored the rat's nest information in the file EXAMPLE:PCrat\_r.

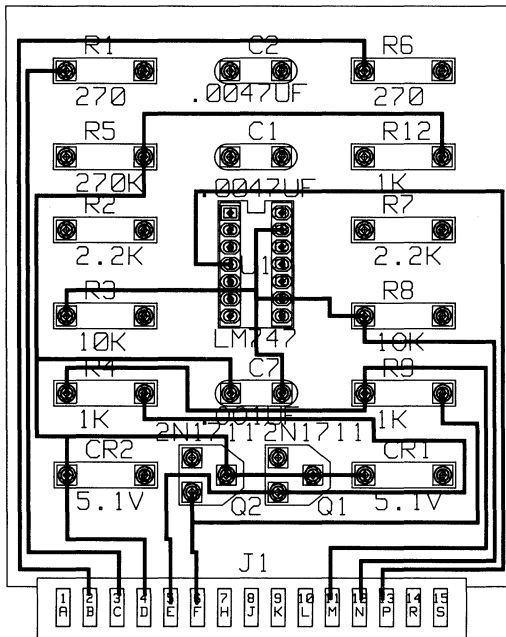
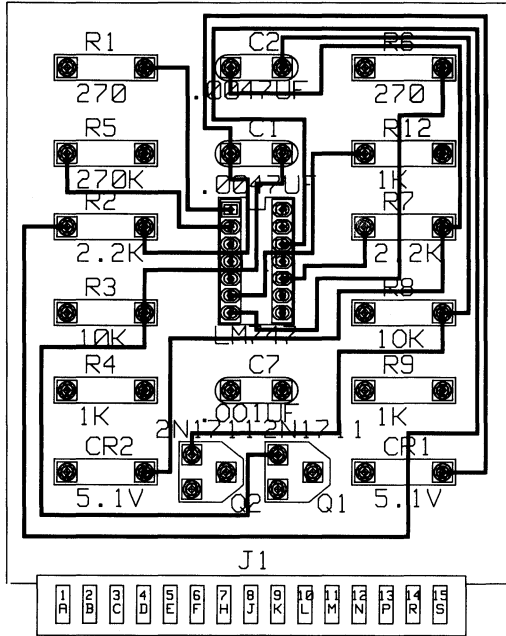
## 6. Press the space bar to return to the Electrical Engineering menu. If necessary correct any errors in the schematic or PC board drawings, and repeat the above procedures.

7. To see the rat's nest information displayed in the PC board drawing, return to the PC Board Layout personality and call up `PCBK_d` using the `EDIT` command.
8. Access the rat's nest file using the `INPUT` command (`INP EXAMPLE:PCrat_r`). The system will begin placing individual lines (traces) point-to-point on the drawing. With all the traces placed on all layers, the drawing looks like this:



9. Save the drawing under a new name (`EXAMPLE:PCroute_d`) so that you can route traces later when convenient.
10. To route the traces, show each layer of traces one at a time (using `SHOW #E -L6;`). One easy way to route traces is to set the lock angle to  $90^\circ$  and use the `STRETCH :T` command in the vertex mode. Note that the macro `ROUTE*` supplied with the PC Board Layout personality performs the same operation.

The two sides of the example PC board (saved in EXAMPLE:PCroute2\_d) might look like this once routed:





## Comparing Connection Lists

The Compare Connection Lists program compares two separate connection lists (such as from a schematic and a PC board) for equality. The connection lists must have matching ports, and the main parts of the corresponding reference designators must match (U1A = U1, U45 = U45, U1A <> U2). Net names are ignored unless a net in connection list #1 has the same net name as a net in connection list #2, but the nets do not match. The output is a table listing the nets in connection list #1 that do not match nets in connection list #2. If a net in connection list #2 does not exactly match any net in connection list #1, every net in connection list #1 is searched for any ports that are in connection list #2, and a list of “partial matches” is created.

When you run this program you are given the options to:

- Ignore single port nets
- Report only net matches
- Report only net mis-matches (errors)
- Report both net matches and net mis-matches

Let’s look at an example. Suppose you want to compare the merged connection list from the schematic drawings (EXAMPLE:merge.c) with a physical connection list of the routed PC board (EXAMPLE:pcroute2.c). You would need to do the following:

1. Generate a connection list from the routed PC board. The resulting output file would be EXAMPLE:pcroute2.c. Any errors would be reported in a file with the same name and the suffix .e.
2. Select the Electrical Engineering option from the main Manager menu, and then select the option named COMPARE CONNECTION LISTS from the next menu.
3. Answer the prompts that appear on the screen as follows:

```
Enter the name of the message file: EWRC:cutils
Enter the name of the first connect list file: EXAMPLE:pcroute2
Enter the name of the next connect list file: EXAMPLE:merge
Enter the name of the file for the results: EXAMPLE:compex
Should nets with only a single port be ignored? (y/n)  y
Should net mis-matches (errors) be reported? (y/n)  y
Should net matches be reported? (y/n)  y
```

The first prompt is answered by the system. You need not type in file name suffixes, but you must type in complete file specifications.

4. Following the last entry, the system processes the information and displays these messages:

```
Processing EXAMPLE:pcroute2_c
Processing EXAMPLE:merge_c and comparing . . . . .
Done,
```

The system has stored the information in the file EXAMPLE:COMPEX, which looks like this:

HP EGS Connect List Comparison

```
Report Matching Nets
Report Nets That Do Not Match
Ignore Nets With Only One Port
```

EXAMPLE:merge_c		EXAMPLE:pcroute2_c
1		15
3		44
4		38
5		32
6		21
8		10
9		28
10		34
11		3
12		25
13		13
14	+15V	19
15	+5V	41
16	-15V	42
17	GND	23
18	MERGE1	43
19	MERGE2	30
20	REFERENCE SIGNAL	8

Connect Lists match exactly.

5. Press the space bar to return to the Electrical Engineering menu. If appropriate, correct any errors in the schematic and PC board drawings, re-run the associated connection lists, and begin the connect list comparison again.

# Photoplot and Drill Post-Processors

## Chapter 5: Photoplot and Drill Post-Processors

Introduction .....	171
Installing the Photoplot and Drill Post-Processors .....	172
Local Installation .....	172
SRM Installation .....	173
Installing Your Codeword .....	174
Obtaining Your Codeword .....	176
The Codeword Installation Program .....	178
Using the Photoplot and Drill Post-Processors .....	179
Operating from the HP EGS Manager Menu .....	179
Operating from the Pascal Command Line .....	180
Overview of PHOTOPLOT .....	181
PHOTOPLOT Output .....	183
Overview of DRILL .....	194
DRILL Output .....	194
Using Tagged Associated Text .....	197
The Wheel File .....	199
The Default File .....	201
Example Default File .....	202
PHOTOPLOT/DRILL Commands .....	203
Customizing the Message File .....	216
Creating Library Parts .....	219
Software Problems .....	225



# Photoplot and Drill Post-Processors

Chapter

5

## Introduction

This chapter describes how to use two HP EGS post-processors to design and fabricate printed circuit boards. These post-processors – called PHOTOPLOT and DRILL – allow you to produce Gerber photoplotter drawing files and Excellon drill files from HP EGS drawing information, which is stored in HP EGS generate files.

The PHOTOPLOT post-processor inputs HP EGS generate files and produces Gerber photoplotter drawing files. Each Gerber drawing file contains the drawing information necessary to produce 1 “film” on the photoplotter.

The DRILL post-processor inputs the same generate files used for PHOTOPLOT and produces Excellon N/C drill files. Each Excellon drill file contains the drawing information necessary to drill 1 set of holes.

PHOTOPLOT and DRILL are contained within the same program for your convenience. Hence on any given run of the program, you may specify which of the following items you want:

- PHOTOPLOT output
- DRILL output
- Attribute data (i.e., a record of input parameters) that includes statistics and other data computed for one particular run of the program

PHOTOPLOT is designed to work only with Gerber-type photoplotters. All photoplotting is done with the following set of Gerber commands: “G01” (linear interpolation), “G54” (aperture select), “G55” (flash mode), “D01” (shutter open), “D02” (shutter close), “D03” (flash), “D10” - “D19”, “D70”, “D71”, “D20” - “D29”, “D72”, “D73” (aperture select), “M00” (program stop), and “M02” (end of program). These codes may be changed literally as you desire (e.g., “D01” may be changed to “ANYCODE”).

DRILL is designed to work with Excellon-format N/C drill machines. DRILL output is a list of tool numbers, followed by list of x,y drill coordinates.

Before reading this chapter and using the PHOTOPLOT and DRILL post-processors, you should have a thorough understanding of HP EGS generate files created with the GENERATE and GENERATE :I commands of the Graphics Editor. The :I option specifies that all implicit steps be changed into explicit steps; implicit step data is ignored by this program. For more information on the GENERATE command, please refer to the *HP EGS Syntax Reference*.

## Installing the Photoplot and Drill Post-Processors

The following sections describe how to install the PHOTOPLOT/DRILL post-processors on either a local disc or a Shared Resource Manager (SRM). Read only the section that pertains to your system. These instructions assume you have previously installed the HP EGS software.

### Local Installation

1. Make sure that the Graphics Editor and the General Drawing Personality are installed on your system in the volumes EWSYS: and EWCODE:. These must be installed before you can use the PHOTOPLOT/DRILL post-processors. To verify you have installed these items, please refer to the verification procedure described in the *Installing HP EGS* manual.
2. Make sure that a volume named EWOPT: exists on your local disc. This is the volume where PHOTOPLOT/DRILL will reside. If no such volume exists, return to the *Installing HP EGS* manual for instructions on creating a volume.
3. Insert the disc labeled PHDR1 in unit number 3. This is the right drive in an HP 9836A/C and the left drive in an HP 9121D or HP 82901A Flexible Disc drive.
4. Starting from the HP EGS Manager menu, use the arrow keys to select System Utilities, and press **Return**.
5. Select the Pascal Command Line from this menu, and press **Return**.
6. Press **S** (for Stream) to initiate the installation procedure.
7. When the system prompts you for the name of the stream file, type in:

```
PHDR1:phdr.lin Return
```

Then follow the instructions displayed on the screen. The system will ask you to insert each disc and press the space bar to continue with the installation procedure. The system will copy a total of 2 discs into the volume EWOPT:. When the procedure is finished, a message indicating successful completion will be displayed.

## SRM Installation

1. Make sure that the Graphics Editor and the General Drawing Personality are installed on your system in the directories EWSYS and EWCODE. These must be installed before you can use the PHOTOPLOT/DRILL post-processors. To verify you have installed these items, please refer to the verification procedure described in the *Installing HP EGS* manual.
2. Make sure that a directory named EWOPT exists on your SRM. If no such directory exists, return to *Installing HP EGS* for instructions first on creating an SRM directory.
3. Insert the disc labeled PHDR1 in unit number 3. This is the right drive in an HP 9836A/C and the left drive in an HP 9121D or HP 82901A Flexible Disc drive.
4. Starting from the HP EGS Manager menu, use the arrow keys to select System Utilities, and press **Return**.
5. Select the Pascal Command Line from this menu, and press **Return**.
6. Press **S** (for Stream) to initiate the installation procedure.
7. When the system asks you for the name of the stream file, type in:

PHDR1:phdr\_sin **Return**

Then follow the instructions displayed on the screen. The system will ask you for the name of the system directory and the optional product directory (e.g., #5:/HP\_EGS). Enter exactly the same information that was entered when you created the directory. The system will ask you to remove each flexible disc after everything is copied, insert the next disc, and press the space bar to continue with the installation procedure. The system will copy a total of 2 discs into the directory EWOPT. When the procedure is finished, a message indicating successful completion will be displayed.

## Installing Your Codeword

The PHOTOPLOT/DRILL program requires a codeword in order to operate. Once you have installed the PHOTOPLOT/DRILL software, you can follow the procedure in this section. One thing to note: If you live in the U.S. or Canada, the following procedure requires you to phone Hewlett-Packard during business hours (Mountain Time). If you live outside the U.S. or Canada, you must contact your HP Systems Engineer to obtain your codeword. It should take you about an hour to obtain and install your codeword.

A codeword is a 16-digit number that you enter once to permanently link your HP EGS software to a computer or HP-HIL ID Module. Codewords are also needed for the HP EGS personalities, the Connection Lister, and the optional IGES Translator. A unique codeword is needed for each workstation using HP EGS. A codeword is also needed for every workstation that operates HP EGS off of an SRM. A codeword is *not* a password that you type in each time you enter HP EGS.

In your HP EGS packaging you received a 7-by-9 inch envelope containing your PHOTOPLOT/DRILL codeword certificate. Do **NOT** open this envelope until you are certain you plan to keep your HP EGS PHOTOPLOT/DRILL software. Once the envelope has been opened, the system **cannot** be returned to HP.



The envelope contains a certificate similar to the one below:



### RIGHT-TO-USE CERTIFICATE FOR HP EGS (98305A)

CERTIFICATE SERIAL NUMBER

Please read this certificate. It tells you what you are entitled to with HP EGS and how to obtain your HP EGS codeword.

1. This certificate provides the serial number (printed above) and instructions you need to obtain your HP EGS codeword. You must have a codeword to save any HP EGS drawings. The codeword links your copy of HP EGS to a particular device (such as a computer or HP-HIL ID Module). As you follow the instructions for obtaining the codeword, remember that your copy of HP EGS can be run only on the device you specify.
2. This certificate entitles you, the purchaser, to use the one (1) copy of HP EGS (98305A) that is packaged with this certificate with one (1) computer or HP-HIL ID Module. You, the purchaser, specify the computer or HP-HIL ID Module when you obtain your HP EGS codeword.
  - This is a non-transferrable license to use 98305A according to Hewlett-Packard's rights and privileges statement. Under no circumstances is this Right-To-Use privilege transferrable.
  - 98305A is copyrighted and may not be copied (except for archive purposes, to replace a defective copy, or for program error verification).
3. To obtain your HP EGS codeword:
  - a. Select "System Utilities" from the HP EGS Manager menu. From the System Utilities menu select "Install Codeword".
 

The program finds the serial number and product number of each device that can be linked to your copy of HP EGS. For additional information on this step, see the section called "Installing Your HP EGS Codeword" in Installing HP EGS 2.1.
  - b. Write down the serial number and product number of the device that you plan to link to your copy of HP EGS.

MODEL \_\_\_\_\_ COMPUTER SERIAL NUMBER \_\_\_\_\_

- c. Obtain your codeword by:
  - Phoning (303) 226-3800 ext. 2222 if you live in the U.S. or Canada. Ask for the FEO Codeword Delivery Service.
  - Contacting your HP Systems Engineer or HP Application Support Representative if you live outside the U.S. or Canada.

The person you contact will ask for the certificate serial number at the top of this sheet and the two numbers you wrote down in the previous step.
- d. Write down the 16-digit codeword issued by the person you contact at

CODEWORD

Hewlett-Packard.

- e. To install your codeword, follow the steps in the section called "Installing Your HP EGS Codeword" of Installing HP EGS 2.1.
4. Retain this certificate for future reference. A Hewlett-Packard representative may need to see it at a later time.

PART NO. 98305-85001

## The HP EGS Right-To-Use Certificate

## Obtaining Your Codeword

1. Look at your certificate and note the serial number at the top.
2. If your computer displays the HP EGS Manager menu,

```

==>System Utilities
    General Drawing
    Electrical Engineering
  
```

skip to step 3. If not, continue.

- a. Turn off your computer (your computer only; do not turn off the disc).
- b. Turn your computer back on. **Hold down the space bar.**
- c. When you see 1E SYSTEM\_EGS at the top right corner of your screen, type in:

```
1E
```

The 1E will appear in the bottom right corner of your computer screen.

- d. The computer displays a screenful of HP EGS copyright information. Press the space bar to continue.
  - e. The computer prompts for the date. Enter the date and press **Return** or simply press **Return** to avoid the prompt.
  - f. The computer prompts for the time. Enter the time and press **Return** or simply press **Return** to avoid the prompt.
3. Select `System Utilities` from the HP EGS Manager Menu. To select a menu item, position the arrow by the menu item with the arrow keys. Press **Return**.
  4. Select `Install Codeword` from the System Utilities Menu. Press **Return**.

This utility runs a program to determine the product and serial numbers of your computer and/or any HP-HIL ID Module(s). You will see a display similar to the one below.

```
CPU ID Prom - Product number : 9836C  Serial number : 2250A00749
```

The CPU ID Prom is the identifier of your computer. In the example above, the product number is 9836C and the serial number is the 10-digit number 2250A00749.

If you have more than one module or both a computer and a module, you must decide which of these you want to link to your copy of PHOTOPLOT/DRILL. After you obtain your codeword you can use your copy of PHOTOPLOT/DRILL only with the computer or HP-HIL ID Module you have chosen. The advantage of linking your copy of PHOTOPLOT/DRILL to a module is that you can take your copy of PHOTOPLOT/DRILL and your module to other workstations.

5. Write down the product and serial numbers of the computer or HP-HIL ID Module that you plan to link with your copy of HP EGS:

Product Number: \_ \_ \_ \_ \_ Serial Number: \_ \_ \_ \_ \_

6. If you live in the U.S. or Canada, phone the number shown in step 3c of your codeword certificate. Ask for the HP EGS Codeword Delivery Service.

If you live outside the U.S. or Canada, contact your HP Systems Engineer or HP Application Support Representative.

7. The person you contact at Hewlett-Packard will ask for the:
  - Certificate serial number. This is listed at the top of your Codeword Certificate.
  - Product number of the device you are going to link to your copy of HP EGS. You wrote this down in step 5.
  - Serial number of the device you are going to link to your copy of HP EGS. You also wrote this down in step 5.

8. The person will then give you your 16-digit codeword. Write it both below and on your certificate.

CODEWORD: \_ \_ \_ \_ \_

Continue to the next section.

## The Codeword Installation Program

To install your HP EGS PHOTOPLOT/DRILL codeword, follow these steps. If you have any problems with the procedure, restart from step 1. To exit this utility, press **Stop**.

1. Continue from the Install Codeword utility explained in the previous section. Press the space bar until a prompt asks if you are ready to install the codeword.
2. The computer prompts you to enter a number between 1 and 4. These numbers represent the function for which you are installing the codeword. Enter:

3

to install the codeword for the optional Photoplotter and N/C drill utilities, and press **Return**.

3. Type in your codeword as prompted. Then press **Return**.
4. The computer displays the name of the message file (`PHDR.MSG,ASC`) in which this utility will place your codeword. Simply press **Return**. Note: If you entered an SRM path name different than the one suggested in this manual, enter this path name in place of the one listed.

The system returns you to the System Utilities menu. To return to the HP EGS Manager menu, select `Return to Main Menu`.

The codeword should now be installed and you can use the PHOTOPLOT/DRILL software. Note that you must install one codeword for each computer on an SRM system.

## Using the Photoplot and Drill Post-Processors

You may run the PHOTOPLOT/DRILL program from either the HP EGS Manager or from the Pascal workstation. The following procedures explain both options.

### Operating from the HP EGS Manager

1. To run the PHOTOPLOT/DRILL post-processors program from the HP EGS Manager, select PHOTOPLOT/DRILL Post-Processors from the HP EGS Electrical Engineering menu. The system will automatically load and execute the PHOTOPLOT/DRILL post-processors program.
2. The system will provide default answers for questions about the error trace file and message file names. The system uses the error trace file to report unexpected errors. It uses the message file to obtain internal data for operation. To change the default name of the error trace file, you must edit the control file EWOPT:phdr\_cn.ASC. More detail on changing a control file appears in Chapter 2: *Customizing Your System*.
3. Once these default file names have been entered by the system and accepted by the PHOTOPLOT/DRILL program, the program will respond with the prompt:

>

PHOTOPLOT/DRILL is now ready to accept commands from you.

4. When you have finished working, you may leave the PHOTOPLOT/DRILL program by entering the QUIT command. This command will return you to the HP EGS main Manager menu.

## Operating from the Pascal Command Line

1. To begin running the PHOTOPLOT/DRILL program from the main Pascal command line, press  to execute a program.
2. When the system asks for the name of the code file, type in:

EWOPT:PHDR  for a local disc

#5:/HP\_EGS/EWOPT/PHDR  for an SRM

If your SRM directory path name differs from the one shown above, type in the correct path name for your system.

3. The program will then ask you for the names of an error trace file and a message file. The system uses the error trace file to report unexpected errors. It uses the message file to obtain internal data for operation.

For the error trace file, type in any file name you wish to use, or PRINTER: if you wish to send the output to a printer. For the message file, type in:

EWOPT:PHDR\_MS,ASC  for a local disc

#5:/HP\_EGS/EWOPT/PHDR\_MS,ASC  for an SRM

If your SRM directory path name differs from the one shown above, be sure to type in the correct path name for your system.

4. Once these file names have been entered and accepted, the program will respond with the prompt:

>

PHOTOPLOT/DRILL is now ready to accept commands from you.

5. When you have finished working, you may leave the PHOTOPLOT/DRILL program by entering the QUIT command. This command will return you to the main Pascal command line.

---

### Note

If you have trouble executing the program, check the message file EWOPT:phdr\_ms.ASC to ensure that the path names for EXEC\_DEF\_FILE (the default file) and EXEC\_FONT\_TABLE (the note stroke table) are correct for your system (SRM or local disc).

---

## Overview of PHOTOPLOT

This section provides an overview of the functionality of the PHOTOPLOT post-processor. It assumes you are familiar with a photoplotter. Throughout this chapter, the term “component” has a specific meaning. “Component” refers to an HP EGS component, which follows the ADD commands in an HP EGS generate file. For example, the term “circle component” refers to the ADD C command, “rectangle component” refers to the ADD R command, and so on. We use the word “component” to avoid confusion between geometric shapes discussed in a general sense and actual HP EGS components.

A photoplotter is a device that produces plots on light sensitive film. The light is directed through specially shaped holes, called apertures. By moving the aperture over the light sensitive film, the photoplotter can draw long lines and other large shapes on that film. Gerber photoplotters are capable of holding 24 apertures in a device known as an aperture wheel. When photoplotting, the photoplotter first spins the wheel to position the correct aperture in front of the light source, and then it begins plotting.

Apertures come in practically any shape. Three aperture shapes are recognized by PHOTOPLOT: circle, square, and rectangle. Apertures may have any shape in a physical aperture wheel. For example, you may wish to use a special registration “target” aperture for aligning several films. PHOTOPLOT simply needs to have them categorized for dimensional (width and height) considerations. Guidelines for doing this are outlined in the *Wheel File* section of this chapter.

A photoplotter can draw in either of two ways: tracing or flashing. PHOTOPLOT recognizes two types of apertures: trace apertures and flash apertures. Trace and flash apertures each use different values of neutral density filters to control light intensity. Thus, PHOTOPLOT requires them to be categorized into the two respective groups.

When tracing, the photoplotter selects an aperture and traces a figure by shining light through the aperture as it moves. If the figure is to be filled, the photoplotter fills it by continually tracing and overlapping traces until all of the figure is filled.

Although tracing provides the desired plot, it is often a very time consuming process. Another method of photoplotting, called flashing, involves the use of specially shaped apertures (flash apertures). For this method, the photoplotter selects the appropriate flash aperture (as instructed by you or automatically by PHOTOPLOT), then moves to the correct position over the drawing, and exposes the film by flashing the light on and off. A copy of the aperture shape is produced on the film. Clearly this is a more time-efficient method of plotting than tracing. However, it requires that apertures be dedicated to specific shapes; the size of these shapes is limited by the physical size of the aperture. Flashing is typically used for recurring features in a drawing or for special shapes which are difficult to draw, such as registration “targets” used to align separate films.

PHOTOPLOT takes as its input HP EGS generate files and reads into memory one portion (as much as will fit in memory) of the first, and perhaps only, generate file. After parsing this data and assuring that it is syntactically correct, it proceeds to sort the HP EGS components according to proximity (if you have specified this sort key). This sorting step assures that the Gerber photoplotter will be given efficient drawing code that minimizes tracing time. Next, the data is sorted by aperture (from smallest to largest aperture number, if you have specified this sort key). This second sort assures that the Gerber photoplotter will not spend an inordinate amount of time switching between apertures while plotting. This sorted or unsorted data is used to produce Gerber plot code. As the Gerber code is produced it is dumped into the output files you specify.

Since several films may be produced in one run of the PHOTOPLOT program, the above process will occur in parallel for all output files you have specified to produce. PHOTOPLOT reads in portions of the generate file and processes them as described above until it reaches the end of all generate files. The size of the “portions” of the HP EGS generate files which are read in one at a time depends only on the amount of memory available on your HP computer and a special memory usage factor. This factor is called EXEC\_MEMORY\_FACTOR and is specified in the message file EWOPT:phdr.ms.ASC.



## PHOTOPLOT Output

You may specify one of two types of PHOTOPLOT output. The first, or “actual” type, is simply Gerber machine-readable code consisting purely of Gerber draw instructions. This type of output is produced when you use the ACTUAL PHOTOPLOT command. The following is a sample of typical Gerber plot code produced using the ACTUAL PHOTOPLOT command.

```
G54D28*
G01X08937Y07222D02*
G55D03*
G01X09012Y07222D02*
G55D03*
G01X09087Y07222D02*
G55D03*
G01X09162Y07222D02*
G55D03*
G54D27*
G01X08942Y07132D02*
G55D03*
G01X09017Y07132D02*
G55D03*
G01X09092Y07132D02*
G55D03*
G01X09167Y07132D02*
G55D03*
G54D26*
G01X09010Y07030D02*
G55D03*
G01X09010Y06915D02*
G55D03*
```

A second type of Gerber output is available to additionally process the PHOTOPLOT output. This second type, or “virtual” PHOTOPLOT output, is identical to “actual” output except that it contains the original (but scaled and translated) generate file commands that PHOTOPLOT used to produce the Gerber plot code. This “virtual” output is obtained using the VIRTUAL PHOTOPLOT command. The output file consists of a list of ADD <component> commands (those seen in a generate file), each followed by the Gerber draw commands (machine-readable code) necessary to produce that HP EGS component on a Gerber photoplotter. An example of a typical VIRTUAL PHOTOPLOT output is shown here.

```

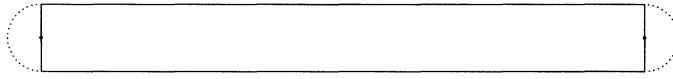
ADD R1 :W0 8910,7190 8965,7255;
G54D28*
G01X08937Y07222D02*
G55D03*
ADD R1 :W0 8985,7190 9040,7255;
G01X09012Y07222D02*
G55D03*
ADD R1 :W0 9060,7190 9115,7255;
G01X09087Y07222D02*
G55D03*
ADD R1 :W0 9135,7190 9190,7255;
G01X09162Y07222D02*
G55D03*
ADD R1 :W0 8910,7105 8975,7160;
G54D27*
G01X08942Y07132D02*
G55D03*
ADD R1 :W0 8985,7105 9050,7160;
G01X09017Y07132D02*
G55D03*
ADD R1 :W0 9060,7105 9125,7160;
G01X09092Y07132D02*
G55D03*
ADD R1 :W0 9135,7105 9200,7160;
G01X09167Y07132D02*
G55D03*
ADD C1 :W0 :R15 9010,7030 9060,7030;
G54D26*
G01X09010Y07030D02*
G55D03*
ADD C1 :W0 :R15 9010,6915 9060,6915;
G01X09010Y06915D02*
G55D03*

```

Only a subset of the HP EGS components are supported by PHOTOPLOT. These components are arcs, circles, hatches, lines, ovals, polygons, rectangles, notes, and text. Dimensions, instances, and markers are not drawn by PHOTOPLOT.

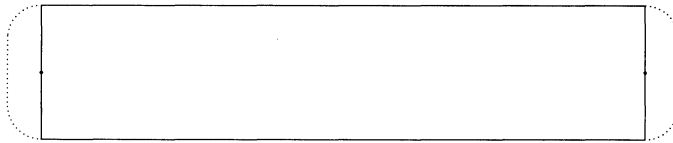
It is possible for a line (trace) to be drawn with an aperture smaller than the width of the line. If this happens, it will take several passes of that aperture to completely fill the line. The following figure shows a case where the line width matches the aperture width. Solid lines indicate the line as it is seen in the Graphics Editor. Dotted lines indicate how the line will be plotted by PHOTOPLOT.

single-pass line



Only one pass of the aperture is required to fill this line. Note the “bumps” on the ends of the line. These bumps are half-circles with radii equal to the width of the circle trace aperture’s radius. The following figure shows a case where the line width exceeds the aperture width.

multi-pass line



This type of line is filled like it were a long, thin rectangle. The “bumps” on the ends appear to be stretched out. Because of overlapping traces, it is possible that “fringing effects,” or hazy edges, can occur from film overexposure. It is difficult to predict when this will occur; it depends on such factors as film used, aperture filter values, line geometry, aperture width, and speed at which the line is traced. (Aperture filters are neutral density filters used to adjust the intensity of the light through the aperture.) In most cases, there is little or no problem with overexposure, given that aperture filter values are selected properly.

The following list explains the transformations that PHOTOPLOT performs:

Notes/Text	Note and text components are considered equivalent forms, and both are drawn as stick figures using, by default, the smallest trace aperture found in the Gerber aperture wheel. You may also override this by using associated text with a tag value of :T1022 (for details, see the section on <i>Using Tagged Associated Text</i> later in this chapter).
Width	All widths on all components are expanded about the centerline of the line segments which make up that component. This is consistent with the HP EGS Graphics Editor.
Line Type	All line types defined in an HP EGS process file (dashed, dotted, etc.) are ignored. All components are drawn with solid lines.
Construction Lines	Construction line (:C) and leader line (:L) options are ignored for line components. All line components are drawn as normal lines.
Filled Lines	Lines are filled using the largest circle trace aperture in the wheel file that is smaller than the width of the line.
Filled Rectangles, Polygons, Hatches	Rectangles, polygons, and hatches are filled using, by default, the smallest circle trace aperture. You may override this by using the POLYGON FILL USE command (described in the <i>PHOTOPLOT/DRILL Commands</i> section later in this chapter). You may also specify the trace aperture by using associated text with a tag value of :T1021 (for details, see the section on <i>Using Tagged Associated Text</i> later in this chapter).
Arcs/Ovals	Arcs and ovals are filled using the smallest trace aperture.
Circles	Circles are filled using the largest circle trace aperture that is smaller than the radius of the circle to be filled.
Hatches	Hatches are filled as solid areas regardless of the type of hatching specified in the generate file (i.e., the :S option is ignored). The NO_FILL :T1024 tag is ignored for this component (for details, see the section on <i>Using Tagged Associated Text</i> later in this chapter).

PHOTOPLOT supports two types of fill modes: normal and FILL WIDTH. In the normal mode only those components which are filled in the HP EGS Graphics Editor (i.e., those with a :FILL option) are filled. In the FILL WIDTH mode, all components (arcs, circles, hatches, lines, polygons, and rectangles) with width will have that width filled, and all components with zero width will be completely filled.

All components which are not filled are drawn using, by default, the smallest circle trace aperture. You may override this by using the NO FILL USE command.

Circle components and rectangle components may be flashed by adding associated text with the tag value :T1020 (for details, see the *Using Associated Tagged Text* section later in this chapter). Flashing has the effect of replacing the circle or rectangle component with the shape of the flash aperture. Only circle components and rectangle components may be flashed. A circle component and a rectangle component may be flashed only by a “flash” or other “special” aperture. It is important to note that the physical shape of apertures need not necessarily be a rectangle or circle. In fact, they may be any shape.

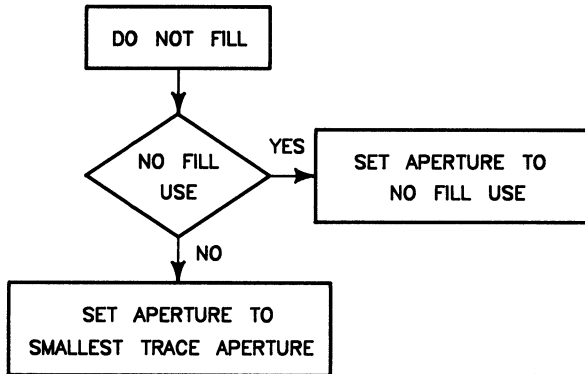
Circles and rectangles can also be flashed using the AUTOFLASH command. This command specifies that PHOTOPLOT is to search the wheel file for flash-type apertures matching each circle and rectangle. If one is found, that aperture is used to flash the related component.

Any component which would be filled (i.e., using the FILL WIDTH mode) can be specified as unfilled by associating the text string “NO\_FILL” with a tag value of :T1024 with that component. Also, associating the text string “IGNORE” with a tag value of :T1024 with any component will cause that component to be totally ignored (not drawn or even seen) by PHOTOPLOT.

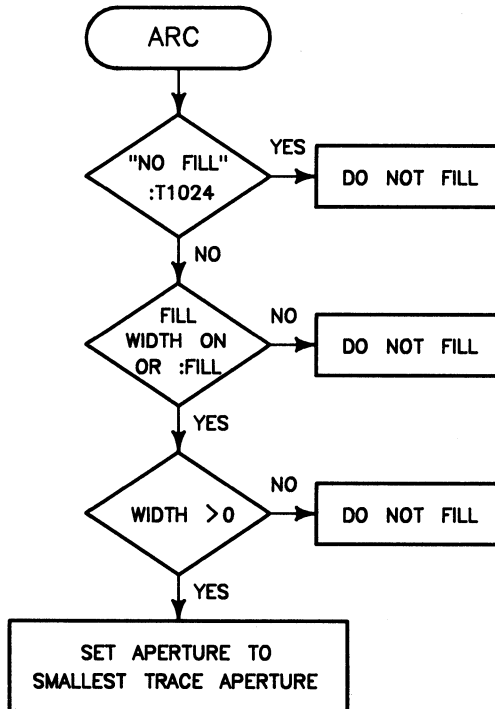
You must specify the photoplot origin using the PHOTOPLOT ORIGIN command. This command allows you to map any point in the HP EGS drawing into any physical photoplotter coordinates. This action offsets the output of the Gerber drawing code. When producing Gerber photoplotter drawing code, PHOTOPLOT positions the photoplot origin at any physical coordinates specified by the PHOTOPLOT ORIGIN command. Thus, you may “move the drawing around” on the photoplotter.

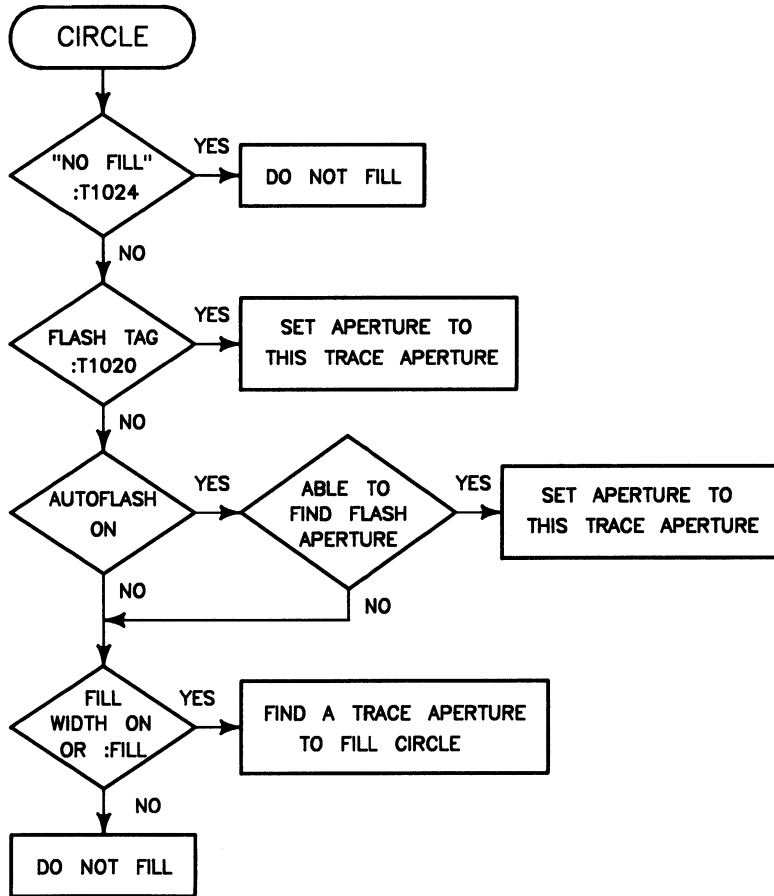
In addition to setting the origin, you may also scale all coordinates and size parameters of an HP EGS component. In effect, you may set a conversion factor (using the PHOTOPLOT UNITS command) to convert between HP EGS drawing units and photoplotter output units if they are not the same.

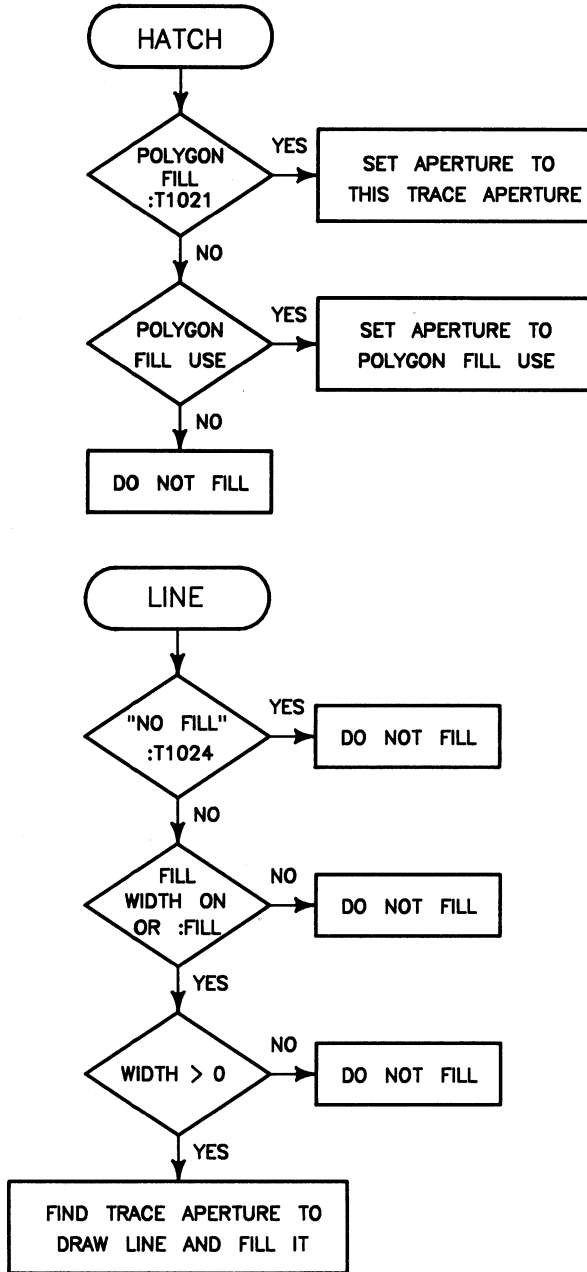
The following flowcharts summarize the rules that PHOTOPLOT uses to select apertures to draw HP EGS components. One reference appears throughout the flowcharts for DO NOT FILL. It indicates:



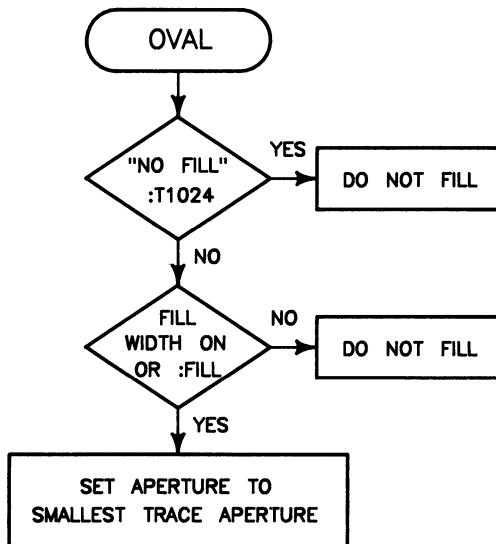
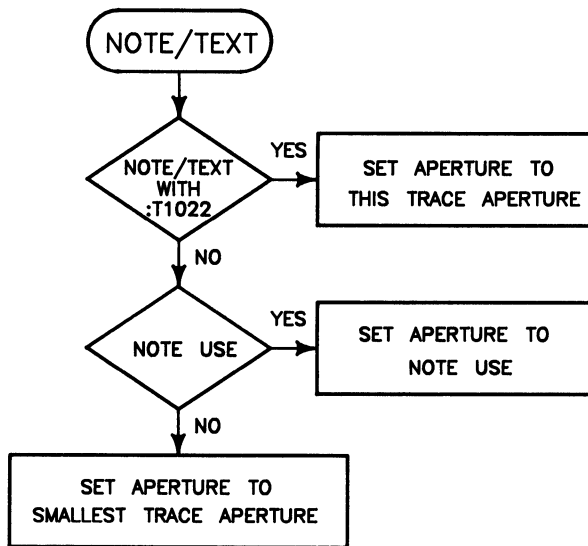
Any references to :FILL indicate the fill option appears in the generate file for that component. The flowcharts for individual components are as follows:

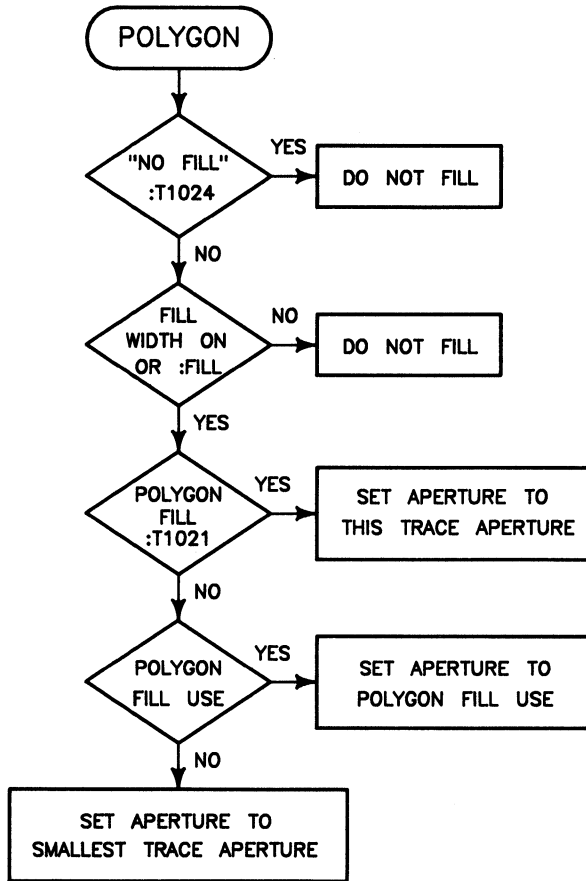


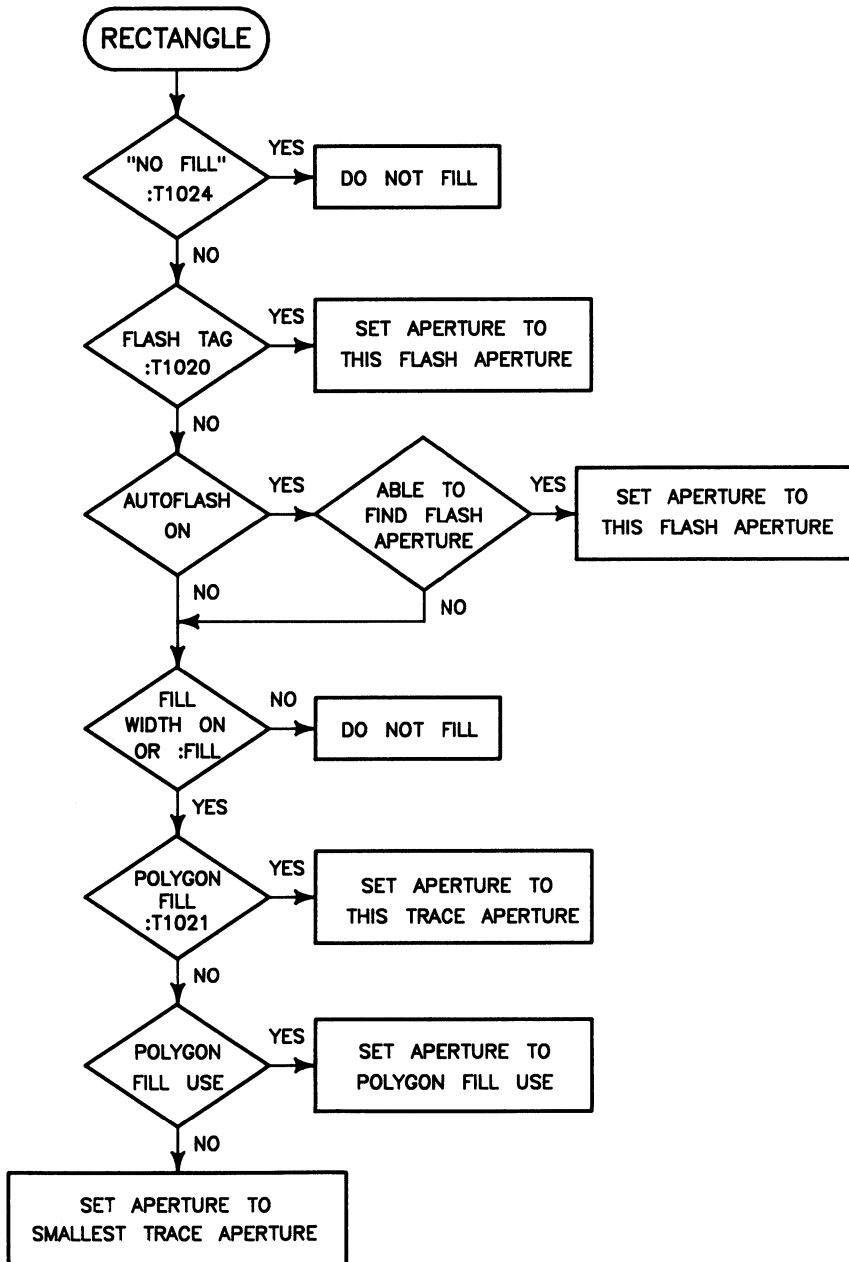












## Overview of DRILL

The DRILL post-processor accepts as its input HP EGS generate files. If used coincidentally with PHOTOPLOT, the set of input generate files is the same. DRILL reads data in a block-by-block fashion concurrent with PHOTOPLOT. DRILL then scans the generate file, searching for specially marked circle components. Any circle tagged with the associated text value of :T1030 is recognized as a hole to be drilled. The finished hole size for a particular hole is the value of the associated text string. If this string is null (blank), the finished hole size is assumed to be the diameter of the circle component.

DRILL then takes this information and produces an Excellon N/C drill output in the user-specified output file. If the DRILL SORT command is used, the output drill holes are sorted by x,y coordinates for each size drill hole (i.e., for each size tool) in the output file. In addition, the hole sizes (or tool numbers) are arranged in order from smallest to largest in each block of each output file. More than one output file may be produced in one pass of DRILL.

You must specify the drill origin. When producing drill coordinates, DRILL positions holes relative to the drill origin in the HP EGS drawing (just as it appears in the HP EGS Graphics Editor). You may position the drill origin at any physical coordinates on the N/C drill machine using the DRILL ORIGIN command. Thus, you may “move the set of drill holes around” to any position on the N/C drill machine.

As with PHOTOPLOT, you may specify a conversion factor (using the DRILL UNITS command) to convert between HP EGS units and DRILL output units.

### DRILL Output

DRILL provides data to drive an Excellon N/C drill machine. DRILL output is a list of tool numbers, with each followed by a list of x,y drill coordinates. The format of the DRILL output is:

```
T(1)           { tool number }
XnnnnnYnnnnn  { drill coordinates }
XnnnnnYnnnnn
T(2)           { tool number }
XnnnnnYnnnnn  { drill coordinates }
.
.
.
M30           { end of output }
```

The characters nnnnn represent actual drill coordinates. M30 specifies the end of the output. Tool numbers T(1), T(2) ..., T(i) are arranged in order such that the size of T(i-1) is less than the size of T(i) (if sorting is specified). The actual tool sizes are given in the attributes file (if one is specified) as well as other data associated with DRILL. An actual DRILL output file would look like this:

```
T01                { tool number }
X10000Y10000      { drill coordinates }
X10000Y11000
T02                { tool number }
X12000Y01000     { drill coordinates }
M30                { end of output }
```

The drill coordinate number format may be changed by using the DRILL FORMAT command. This number is left padded with zeros.

You may specify either of two types of DRILL output. The first, "actual" type, is shown in the above example and is obtained using the ACTUAL DRILL command. The second, called a "virtual" DRILL output, is obtained using the VIRTUAL DRILL command. This second type of output file is provided for those users who desire to additionally process the DRILL output. The format for this second type is shown here:

```
finished hole size { tool size in user units }
x,y coordinate     { drill coordinates }
x,y coordinate
,
,
,
##EOT##            { end of tool marker }
finished hole size { tool size in user units }
x,y coordinate     { drill coordinates }
,
,
,
##EOT##            { end of tool marker }
finished hole size { tool size in user units }
x,y coordinate     { drill coordinates }
,
,
,
##EOT##            { end of tool marker }
```

This type of DRILL output file contains a list of tool sizes (in user units) followed by a list of drill coordinates (in user units). Each tool entry ends with an “end-of-tool” marker. For example, a VIRTUAL DRILL output file would look like this:

```

4,00000E+001           { tool size in user units }
1,00000E+001,7,20000E+001 { drill coordinates }
1,00000E+001,1,05000E+002
1,00000E+001,6,93000E+002
1,04000E+003,6,22200E+003
5,00000E+003,5,00000E+003
##EDT##               { end of tool marker }
5,50000E+001           { tool size in user units }
1,50000E+001,7,20000E+001 { drill coordinates }
1,50000E+001,1,05000E+002
##EDT##               { end of tool marker }
1,19000E+002           { tool size in user units }
8,50000E+001,8,50000E+001 { drill coordinates }
1,00000E+002,1,00000E+002
##EDT##               { end of tool marker }

```

## Using Tagged Associated Text

PHOTOPLOT and DRILL both take advantage of an HP EGS feature known as associated text tag values. Tag values are specified by :Txxxx where xxxx is a user-defined number between 0000 and 9999; these values are associated to components using the ADD TEXT/NOTE command of the Graphics Editor. Several specific tag values are used by PHOTOPLOT and DRILL to identify special information when processing the input generate file.

Note that no more than one specially tagged piece of associated text may be used per component, or else an error will result when PHOTOPLOT/DRILL is run. In addition, only the first level of associated text is recognized by PHOTOPLOT/DRILL. For example, associated text which is associated to another associated text component (i.e., two levels of association) is not recognized.

The special associated text tags used by PHOTOPLOT and DRILL are:

### **:T1020**

Aperture name of a specific aperture in the wheel file (which describes the Gerber aperture wheel) to use in flashing the tagged component. An error will result if this tag is used for any components other than circles and rectangles. If the aperture cannot be found in the wheel file, an error will result. If the aperture is not a “flash” or “special” type, an error will result. For example, a typical generate file segment would appear as the following if a particular tagged circle were to be flashed by a particular aperture.

```
ADD C10 10,10 10,20 ;
BEGIN ASSOC
  ADD N20 :T1020 'circle10' 10,30 ;
END ASSOC
```

This example assumes that the wheel file contains a “flash” aperture or “special” aperture named “circle10”.

### **:T1021**

Aperture name of a specific aperture in the wheel file to use in filling a particular polygon, hatch, or rectangle. If a polygon, hatch, or rectangle component is not tagged with :T1021 and is to be filled, PHOTOPLOT will use either the smallest circle trace aperture (by default) or the aperture specified by the POLYGON FILL USE command, depending on which mode is set. An error will result if this tag is used for any components other than polygons, hatches, and rectangles. If the aperture name cannot be found in the wheel file, an error will result. Apertures must be of type “trace” or else an error will result. This associated text tag does not imply that the component will always be filled; it specifies only that if a component is to be filled, this aperture will be used.

**:T1022**

Aperture name of a specific aperture in the wheel file to use in drawing notes and text. To use this feature, a note or text component must have an accompanying associated text string with the tag value :T1022 specifying the name of the aperture to be used. If this feature is not used, then either the smallest circle trace aperture will be used (by default) or the aperture specified by the NOTE USE command, depending on which mode is set. If this tag is used for any component other than a note or text, an error will result. If this aperture cannot be found in the wheel file, an error will result. Apertures must be of type “trace” or else an error will result.

**:T1024**

IGNORE and NO\_FILL identifier. If the associated text string “IGNORE” with tag value :T1024 is used, the associated component will be ignored by PHOTOPLOT and DRILL. If the associated text string “NO\_FILL” with tag value :T1024 is used, the associated component will not be filled (if it otherwise would have been filled).

**:T1030**

Finished hole size for DRILL. This value tags those circle components to be used as drill holes. The text is the finished hole size in any units (since DRILL does not need to know what units you are using for drilling). If the string is null (blank) or an asterisk \*, the finished hole size will be the diameter of the circle component. The units you use should be consistent from drill hole to drill hole. The text must be a valid numeric string or a null string, or else an error will result. An error will also result if this tag is used for any components other than circles.

**:T1036**

This value tags marker components which are to be recognized as data points by PHOTOPLOT. At run time, PHOTOPLOT will return the real coordinates of each special marker appearing in the HP EGS drawing. The user-specified units for PHOTOPLOT will be used. An error will result if this tag is used for components other than markers.



## The Wheel File

Before using PHOTOPLOT you must first specify the attributes of the Gerber photoplotter's aperture wheel. You should create a wheel file that describes one aperture wheel which will reside on the photoplotter when the resulting photoplots are drawn. You may create many wheel files (using the Pascal Editor) and keep them on-line, but only one wheel file may be used for each PHOTOPLOT run.

A wheel file consists of a list of aperture entries. Any number of aperture entries may exist in a wheel file. An aperture entry must have the following syntax:

```
aperture_name  aperture_number  shape      type      width  [ height ] ;
```

where the following definitions apply.

### **aperture\_name:**

User-defined name (less than or equal to 255 characters) to be associated with a particular aperture entry. This name allows you to refer to a particular aperture for various PHOTOPLOT features.

### **aperture\_number:**

Aperture number of a particular aperture on the aperture wheel. This is the aperture number appearing in the Gerber output file. For example, 10 could be the number of the first Gerber aperture, 11 could be the second, and so on.

### **shape:**

The shape of an aperture. Three shapes are allowed: circle, square, and rectangle. Physical aperture shapes which are not one of the three allowable types should use the type which most closely matches the dimensionality of that aperture. For example, any apertures which have unequal widths and heights should use the "rectangle" shape with width and height parameters that closely match those of the irregular aperture shape (so that it is "logically" bordered by such a rectangle). Thus, when PHOTOPLOT uses this aperture it will assume that the center of the rectangle is the center of the aperture, disregarding the actual physical shape of the aperture in the aperture wheel.

**type:**

The aperture type. Three types are allowed: trace, flash, and special. Trace apertures are used for moving draws such as drawing lines and filling areas. Flash apertures are used for flashing pre-defined shapes such as circles and rectangles. Special apertures are used for flashing unusual shapes like company logos or registration targets. PHOTOPLOT uses only flash apertures when searching the wheel file in the AUTOFLASH mode. Only flash and special apertures may be flashed by the :T1020 tagged associated text option.

**width:**

Width or horizontal dimension (in units specified by the UNITS command of the Graphics Editor) of circular, square, and rectangular shapes.

**height:**

Height or vertical dimension (in units specified by UNITS command of the Graphics Editor) of rectangular apertures. Circular and square apertures do not need a specified height. If one is specified for a circle or square, an error will result.

Aperture entries must be terminated by a semicolon. Any syntax errors or parameter errors in a wheel file will be trapped when you use the WHEEL command. Below is an example of a typical wheel file.

```
tr5 10 circle trace 5 ;
tr10 12 circle trace 10;
fl10 19 square flash 10;
tr20 70 circle trace 20 ;
spec1 73 rectangle special 20 30;
```

The wheel file to be used for a particular PHOTOPLOT run is specified by the WHEEL command. If the specified wheel file cannot be found when you instruct PHOTOPLOT to begin processing, or if a wheel file is not specified, a fatal error will result and PHOTOPLOT will terminate that run.

## The Default File

Before using either PHOTOPLOT or DRILL you must also create a special input file known as the default file. This file specifies a set of PHOTOPLOT/DRILL program parameters to be used for a particular run. As the name implies, these are simply a set of default parameter values that you may wish to keep together under one single name and use again and again.

The default file contains a number of commands that end with an optional semicolon. Commands perform a number of different actions such as setting modes, specifying numeric data, specifying file names to be used, and so on. In fact, all parameters of a particular run of PHOTOPLOT/DRILL may be completely specified by the commands in a default file. In essence, a default file is nothing more than a stream file that you input (using the INPUT command) from the keyboard. All commands may appear either in a default file or be typed in directly from the keyboard. To protect the user, the RESET and QUIT commands are ignored by the INPUT command.

Since default files reside in the Pascal file system, you may create any number of them and specify which one you want to use for a particular PHOTOPLOT/DRILL run. If no default file is input, the program will use a set of pre-programmed default values. If a default file is input, you may still opt to modify some of the program parameters for a particular run. You may edit the parameter values one by one by typing in the appropriate commands. In fact, you may even input another default file and start over again if by chance you make a mistake. When all parameter values are set as you wish, you should instruct the program to start processing using the BEGIN command.

When the program is started or a RESET command is executed, the system executes an `INPUT EWOPT:phdr_def` command to call up a special default file named `phdr_def`, which is supplied with the system. If this file cannot be found, you will be informed of an error; you may continue to run the program as the system will use its pre-programmed default values. Because `phdr_def` is like any other default file, you may edit it as desired to provide your own unique default environment at startup time and after resets. In addition, you may change the default file name in the message file `EWOPT:phdr_ms.ASC` to point to any default file.

## Example Default File

The following is an example of a typical default file. A detailed explanation of each command appears in the *PHOTOPLOT/DRILL Commands* section of this chapter.

```

ACTUAL PHOTOPLOT film1 2,3,4,5 ;
ACTUAL PHOTOPLOT film2 6,7,8,9 ;
ACTUAL DRILL plated 10,11 ;
ACTUAL DRILL unplated 13 ;
SORT APERTURE ON;
SORT PROXIMITY 1000,1000;
SORT DRILL X;
PHOTOPLOT ORIGIN 0,0 2000,2000;
DRILL ORIGIN 0,0 0,0;
POLYGON FILL USE trace10;
NOTE USE trace15;
NO FILL USE trace10;
PHOTOPLOT FORMAT 2,3;
DRILL FORMAT 4,2;
AUTOFLASH ON;
WHEEL mywheel;
PAUSE ON ERROR;

```

In this example, two PHOTOPLOT outputs are to be created. The first output, “film1”, is to be produced from data on layers 2, 3, 4, and 5. The second output, “film2”, is to be produced from data on layers 6, 7, 8, and 9.

Two DRILL outputs are to be created. The first “plated” output is to be produced from data only on layers 10 and 11. The second “unplated” output is to be produced from data on layer 13.

Both types of PHOTOPLOT sorting are specified: by aperture and by proximity. Proximity sorting is to be performed with a cell size of 1000-by-1000 user units. Sorting of DRILL output files by x-coordinate is specified.

PHOTOPLOT and DRILL origins (0,0) are specified at 2000,2000 and 0,0, respectively.

The polygon, hatch, and rectangle default fill aperture is “trace10”. The note and text component default draw aperture is “trace15”. Unfilled components are to be drawn with aperture “trace10”.

The coordinate number format for the emitted Gerber code is 2.3 (2 digits for the whole number part and 3 digits for the fractional part). The number format for the coordinates emitted by DRILL 4.2 (4 digits for the whole number part and 2 digits for the fractional part).

The automatic flash mode is set.

The specified wheel file is “mywheel”.

PHOTOPLOT and DRILL have been instructed to pause and prompt the user if an error occurs.

## PHOTOPLOT/DRILL Commands

This section describes each command you may use in the PHOTOPLOT/DRILL program. Logically related commands are listed together.

These commands are only meaningful to the PHOTOPLOT/DRILL post-processors, and should not be confused with HP EGS Graphics Editor commands. As with HP EGS Graphics Editor commands, you need only use the first three or four letters of any command keyword. In many cases fewer characters may be specified, since the program can figure out what you have input. The semicolons appearing at the end of each command are optional. If ever a command is ambiguous, the system will inform you. Note that the PHOTOPLOT/DRILL program does not distinguish between upper and lower case alpha characters (except for literal strings such as file specifiers, aperture names, etc.).

> **ATTRIBUTES file\_specifier ;**

> **ATTRIBUTES CLEAR;**

This command allows you to specify an attributes file. “File\_specifier” is any legal Pascal file name. The attributes file is an output file which contains vital data and statistics about a particular run of PHOTOPLOT/DRILL. Only one attributes file may be specified. If more than one is specified, the most recent will be used. ATTRIBUTES CLEAR indicates that no attributes file is to be produced. The following is an example of a typical attributes file.

```
*****
HP EGS PHOTOPLOT/DRILL POSTPROCESSOR
10:24:20          9/17/84

AUTOFLASH:       ON
ERROR MODE:      CONTINUE ON ERROR
FILL WIDTH MODE: OFF

SORT DRILL:      MAJOR SORT ON X
SORT PROXIMITY:  X: 1.00000E+001      Y: 1.00000E+001
SORT APERTURE:   ON

DRILL FORMAT:    4.2
PHOTOPLOT FORMAT: 5.0
DR ORIGIN SOURCE: 0.000E+000,0.000E+000  OUTPUT: 1.000E+002,1.000E+002
PH ORIGIN SOURCE: -6.49E+003,-5.63E+003  OUTPUT: 1.000E+003,1.000E+003
```

## 204 Photoplot and Drill Post-Processors

DRILL UNITS FACTOR: 1.000E+000  
PHOTOPLOT UNITS FACTOR: 1.000E+000  
PHOTOPLOT RESOLUTION: OFF

NO FILL USE: none specified  
NOTE USE: T15  
POLYGON FILL USE: T50

ATTRIBUTES FILE: #5:/HP\_EGS/NC/TESTS/PPtest\_attr  
WHEEL FILE: #5:/HP\_EGS/NC/fsd\_wheel

CURRENT HP EGS GENERATE SOURCE FILES:  
#5:/HP\_EGS/NC/TESTS/PPtest\_g

VIRTUAL DRILL FILE:  
#5:/HP\_EGS/NC/TESTS/PPtest\_vdr  
1 2 3 4 5 6 7 8 9 10 100

ACTUAL DRILL FILE:  
#5:/HP\_EGS/NC/TESTS/PPtest\_adr  
1 2 3 4 5 6 7 8 9 10 100

VIRTUAL PHOTOPLOT FILE:  
#5:/HP\_EGS/NC/TESTS/PPtest\_vph  
1 2 3 4 5 6 7 8 9 10

ACTUAL PHOTOPLOT FILE:  
#5:/HP\_EGS/NC/TESTS/PPtest\_aph  
1 2 3 4 5 6 7 8 9 10

ERROR: This type of component is not used by this program.  
DIMENSION component at xy-coordinates:  
2.54000E+003 2.00000E+001

PHOTOPLOT DATA POINT: THIS IS A TEST MARKER  
9.245E+003 7.250E+003

ERROR: Expected NO\_FILL or IGNORE for :T1024.  
RECTANGLE component at xy-coordinates:  
1.89000E+003 3.60000E+002

## TOOL LIST

tool number	count	diameter
-------------	-------	----------

1	2	1.000E+002
2	1	5.000E+001

## WHEEL FILE DATA

aperture name

aperture number	count	type	shape	width	height
-----------------	-------	------	-------	-------	--------

T8	10	25	TRACE	CIRCLE	8.000E+000	0.000E+000
T10	11	0	TRACE	CIRCLE	1.000E+001	0.000E+000
T12	12	0	TRACE	CIRCLE	1.200E+001	0.000E+000
T15	13	14	TRACE	CIRCLE	1.500E+001	0.000E+000
T25	14	1	TRACE	CIRCLE	2.500E+001	0.000E+000
T50	15	6	TRACE	CIRCLE	5.000E+001	0.000E+000
T100	16	4	TRACE	CIRCLE	1.000E+002	0.000E+000
FC50	17	0	FLASH	CIRCLE	5.000E+001	0.000E+000
FC55	18	0	FLASH	CIRCLE	5.500E+001	0.000E+000
FS55	19	0	FLASH	SQUARE	5.500E+001	0.000E+000
FC60	70	0	FLASH	CIRCLE	6.000E+001	0.000E+000
FC68	71	0	FLASH	CIRCLE	6.800E+001	0.000E+000
FS70	20	0	FLASH	SQUARE	7.000E+001	0.000E+000
FC75	21	0	FLASH	CIRCLE	7.500E+001	0.000E+000
FC80	22	0	FLASH	CIRCLE	8.000E+001	0.000E+000
FC85	23	0	FLASH	CIRCLE	8.500E+001	0.000E+000

FC90	24	0	FLASH	CIRCLE	9.000E+001	0.000E+000
FC95	25	0	FLASH	CIRCLE	9.500E+001	0.000E+000
FC100	26	2	FLASH	CIRCLE	1.000E+002	0.000E+000
FO6555	27	4	FLASH	RECTANGLE_H	6.500E+001	5.500E+001
FO6555	28	4	FLASH	RECTANGLE_V	5.500E+001	6.500E+001
FO8070	29	0	FLASH	RECTANGLE_H	8.000E+001	7.000E+001
FO8070	72	0	FLASH	RECTANGLE_V	7.000E+001	8.000E+001
FTARGET	73	0	SPECIAL	SQUARE	1.000E+000	0.000E+000

2 user errors in this run.

> AUTOFLASH ON ;  
 > AUTOFLASH OFF ;

These commands specify one of two flash modes for PHOTOPLOT. AUTOFLASH OFF disables the automatic flash mode, and AUTOFLASH ON enables the automatic flash mode. AUTOFLASH ON instructs PHOTOPLOT to search the wheel file for a “flash” aperture matching the dimensions of any circle or rectangle component in the HP EGS generate file. If this mode is used and a matching flash aperture is not found in the wheel file, the component is filled using an appropriate trace aperture (if that component is to be filled). Only those apertures of type “flash” are examined while performing an AUTOFLASH search in the wheel file. If this mode is not used, you must explicitly specify that a component is to be flashed by associating text with the circle or rectangle to be flashed. The text string must be the aperture name of the desired aperture in the wheel file, and it must have a tag value of :T1020.

Some flash apertures may be used over and over again in many printed circuit board drawings. For example, IC pad flash apertures can occur hundreds of times on a typical board. It may be convenient to use the AUTOFLASH feature of PHOTOPLOT rather than flash tags (:T1020) in the following way. Suppose you have an aperture “PAD” which is specially suited to flash pads for IC sockets. One way to use this aperture is to use a flash tag :T1020. However, you may use AUTOFLASH if the rectangles or circles you want to flash uniquely match the dimensions of apertures found in the wheel file. You should use uniquely dimensioned rectangles (i.e., 13.2 mils by 8.1 mils) which you are certain will never be used to represent an actual rectangle shape. Using this feature can save some memory, and will make generate files much smaller since the associated text information is not required.



**> BEGIN ;**

This command instructs the PHOTOPLOT/DRILL program to start processing as instructed. A wheel file must be specified before this command is used, or an error will result.

**> CONTINUE ON ERROR ;****> PAUSE ON ERROR ;**

These commands specify one of two error reporting modes. PAUSE ON ERROR mode will temporarily pause when a non-fatal error is encountered. This mode will identify the error, and then ask if you wish to proceed. CONTINUE ON ERROR will continue processing if a non-fatal error occurs, signal that an error has occurred, and log it on your terminal. Both modes will log errors in the attributes file (if one is specified).

**> ACTUAL DRILL file\_specifier layer\_list ;****> ACTUAL DRILL CLEAR;****> ACTUAL PHOTOPLOT file\_specifier layer\_list ;****> ACTUAL PHOTOPLOT CLEAR;**

These commands specify output file names for PHOTOPLOT and DRILL, respectively. “File\_specifier” is any legal Pascal file name. These commands also specify what HP EGS layers are associated with each respective output file. “Layer\_list” is simply a list of HP EGS layer numbers (separated by a comma or blank). Hence, there may be multiple occurrences of PHOTOPLOT and DRILL commands in a default file. The following are legal PHOTOPLOT and DRILL commands.

```
ACTUAL DRILL #5:/MYDIR/unPlated 13,14 ;
ACTUAL DRILL #13:Plated 10,11 ;
ACTUAL PHOTOPLOT #12:my_Plot1 2,4,5 6,7 12 156 ;
ACTUAL PHOTOPLOT mask1 34,56 ;
```

The first one assumes that the system has an SRM on line. PHOTOPLOT and DRILL layer lists do not have to be disjoint. In other words, an HP EGS layer number may appear in many layer lists for any given run of PHOTOPLOT/DRILL.

ACTUAL PHOTOPLOT CLEAR or ACTUAL DRILL CLEAR specifies that the effects of previous PHOTOPLOT and DRILL commands are to be ignored (this has the effect of “starting from a clean slate”). These commands produce “actual” type output files as opposed to “virtual” type output files. If you have trouble specifying an output file on a local disc, you are probably out of space. Please refer to the section on *Software Problems* later in this chapter.

> **DRILL FORMAT j.k ;**

> **PHOTOPLOT FORMAT m.n ;**

The DRILL FORMAT command allows you to specify the format of the drill coordinates emitted by DRILL. The j,k parameter specifies to use j digit places to represent the whole number part of the drill coordinate and k digit places to represent the fractional part of the drill coordinate. For example,

```
DRILL FORMAT 4,2 ;
```

would mean that drill coordinates would take the form XjijjkkYjjjkk in the DRILL output file. The numbers are padded with zeros. This number format is also used for tool sizes and drill coordinates when the “virtual” output is specified, except that padded zeros are removed and a decimal point is inserted at the proper position.

The PHOTOPLOT FORMAT command allows you to select the decimal format of the resulting Gerber draw code. The “m” parameter specifies the number of places to hold for the whole number part of a Gerber draw coordinate. The “n” parameter specifies the number of decimal places to hold for the fractional part of a Gerber draw coordinate. For example,

```
PHOTOPLOT FORMAT 2,3 ;  
PHOTOPLOT FORMAT 3,3 ;
```

are valid commands. The latter would allow 3 places for the whole number part of the coordinate and 3 places for the fractional part. It would produce Gerber coordinates of the form XmmmmnnYmmmmnn . Numbers are padded with zeros. If a coordinate cannot be represented in the specified format, an error will result.

> **FILL WIDTH ON ;**

> **FILL WIDTH OFF ;**

These commands specify one of two possible fill modes. If FILL WIDTH OFF is used, normal fill mode is assumed and HP EGS components created with the :FILL option are filled. FILL WIDTH ON specifies that all components (arcs, circles, hatches, lines, polygons and rectangles) with width will have that width filled, and all components with zero width will be completely filled. In the FILL WIDTH ON mode, you may specify :FILL, but this is redundant. In either mode you may specify to not fill a component by associating the text string “NO\_FILL” with that particular component.

- > **HELP ;**
- > **HELP command\_name ;**
- > **HELP CATALOG**

This command gives on-line explanations of each command and other information. HELP <command\_name> explains a specific command. HELP gives general information about the use and function of PHOTOPLOT/DRILL. HELP CATALOG lists all available HELP subjects. Examples are:

```
HELP SORT ;
HELP ;
HELP PHOTOPLOT ;
```

- > **INPUT file\_specifier ;**
- > **INPUT file\_specifier Y;**
- > **INPUT file\_specifier N;**

This command allows you to input a stream file which contains legal commands. This command is used to input default files. "File\_specifier" is any legal Pascal file name. The INPUT command contained in an input file will cause the current input file to be closed; the system will then input the file specified by the new INPUT command. This command may also contain a Y (Yes) or N (No) option to automatically answer all Yes/No questions.

- > **NO FILL USE aperture\_name ;**
- > **NO FILL USE CLEAR;**

This command allows you to specify the aperture to be used to draw unfilled components. If NO FILL USE is not present, by default the smallest circle trace aperture is used. This feature may be useful when an unusually small aperture exists in the wheel file for a special purpose, and you do not want to use such a small aperture to draw the outlines of components. The "aperture\_name" tells which aperture in the wheel file to use to trace the unfilled components. If this aperture cannot be found in the wheel file, and if this aperture is not of type "trace", an error will result. NO FILL USE CLEAR specifies that this mode is to be cancelled, and to use the default mode.

- > **NOTE USE aperture\_name ;**
- > **NOTE USE CLEAR;**

By default, PHOTOPLOT uses the smallest trace aperture in the wheel file to draw note and text components. This command allows you to specify a particular aperture for notes and text. This may be useful where an unusually small trace aperture exists in the wheel file for a particular purpose, and you do not want to draw notes and text with this aperture. If "aperture\_name" is not found in the wheel file, an error will result. If this mode is not used and a trace aperture does not appear in the wheel file, an error will result. An aperture must be of type "trace", or else an error will result. NOTE USE CLEAR specifies that this mode is to be cancelled, and to use the default mode.

> **DRILL ORIGIN** *x , y i , j* ;  
 > **PHOTOPLOT ORIGIN** *x , y i , j* ;

These commands allow you to specify the positions of the PHOTOPLOT output and DRILL output on a photoplotter and N/C drill machine, respectively. DRILL ORIGIN and PHOTOPLOT ORIGIN are functionally equivalent. PHOTOPLOT ORIGIN allows you to specify the mapping between HP EGS drawing coordinates *x,y* and Gerber photoplotter coordinates *i,j*. This command positions the HP EGS drawing coordinate *x,y* at position *i,j* on the photoplotter. Hence, you may offset the photoplotted drawing (“move it around”) to any desired position on the photoplotter. DRILL ORIGIN allows you to specify the mapping between HP EGS drawing coordinates *x,y* and N/C drill coordinates *i,j*. This command specifies the position of the HP EGS drawing coordinates *x,y* at position *i,j* on the N/C drill machine. Hence, you may offset a set of drill holes (“move them around”) to any desired position on the N/C drill machine. For example, the command

```
PHOTOPLOT ORIGIN 0,0 5,10 ;
```

would position the HP EGS drawing origin (0,0) at 5,10 on the Gerber photoplotter (and of course, shift everything else in the drawing accordingly). The coordinates *x,y* are in the user units used in the HP EGS drawing. The Gerber photoplotter coordinates *i,j* are in the units used in the PHOTOPLOT output code (set by the PHOTOPLOT UNITS FACTOR command). The N/C drill coordinates *i,j* are in the units used in the DRILL output code (set by the DRILL UNITS FACTOR command). The PHOTOPLOT ORIGIN and DRILL ORIGIN commands know about the PHOTOPLOT UNITS FACTOR command and the DRILL UNITS FACTOR command, and will make the proper internal units conversions.

> **POLYGON FILL USE** *aperture\_name* ;  
 > **POLYGON FILL USE CLEAR**;

Normally when filling polygons, hatches, and rectangles PHOTOPLOT uses, by default, the smallest circle trace aperture found in the wheel file. This command allows you to specify a particular trace aperture to fill polygons, hatches, and rectangles. This allows experienced users to speed up such fills when high precision (i.e., the smallest aperture) is not required. “Aperture\_name” specifies the particular aperture in the wheel file to use. If it is not found in the wheel file, or the aperture is not “trace” type, an error will result. If this mode is not specified, the system will default to the smallest trace aperture (if the component is to be filled). POLYGON FILL USE CLEAR specifies that this mode is to be cancelled, and to use the default mode.

> **PREFIX** *string* ;  
 > **PREFIX CLEAR** ;

This command allows you to set a local prefix (local only to this program) as a typing aid.

This *does not* change or in any way affect the current HP EGS prefix. If any file name does not contain a “:” and a prefix is currently set, the program will append the prefix to the beginning of the file name. For SRM prefixes, the last slash mark is optional. For example, the commands

```
PREFIX #5:/MYVOL/
INPUT p_hdr_def
```

would first set the prefix to #5:/MYVOL/, then stream the input file `p_hdr_def` from volume MYVOL:. In effect, this is equivalent to

```
INPUT #5:/MYVOL/p_hdr_def
```

If you do not wish the prefix to be appended to the beginning of a file name (and a prefix is currently specified), simply specify that file name with a leading “:”. For example, in the command

```
inp :p_hdr_def
```

the “:” would specify not to add the prefix to the file name `p_hdr_def`. In such cases the current HP EGS prefix would be used. This is the prefix set when PHOTOPLOT/DRILL was started up.

PREFIX CLEAR clears the current PHOTOPLOT/DRILL prefix.

#### > RESET ;

This command tells PHOTOPLOT/DRILL to “forget” the effects of all previous commands and re-initialize as though the program just started up.

#### > RESOLUTION resolution ;

#### > RESOLUTION CLEAR ;

You may override the resolution specified for arcs and circles with this command. If used, PHOTOPLOT will ignore the resolution specified in the HP EGS generate file for all arcs and circles, and will use instead the value specified by “resolution”. “Resolution” is an angular measure given in degrees. The minimum resolution PHOTOPLOT will allow is 1.0 degrees, and the maximum is 120.0 degrees. However, PHOTOPLOT can be instructed to determine the maximum smoothness possible for arcs and circles by specifying a resolution of 0. The following are examples of the RESOLUTION command:

```
RESOLUTION 20 ;
RESOLUTION 1.5 ;
RESOLUTION 0 ;
```

RESOLUTION CLEAR cancels the override and specifies that the generate file resolutions be used.

- > **SORT APERTURE ON ;**
- > **SORT APERTURE OFF ;**
- > **SORT PROXIMITY horizontal\_quantization , vertical\_quantization ;**
- > **SORT PROXIMITY OFF ;**

These commands specify whether the sort capabilities are enabled or disabled for PHOTOPLOT. SORT APERTURE ON sorts the Gerber photoplotter data according to aperture. Apertures are sorted by aperture number. SORT APERTURE OFF turns this aperture sort off. SORT PROXIMITY with horizontal and vertical quantization values specified will sort by grouping objects which are close together within the same cells. Horizontal\_quantization and vertical\_quantization set the imaginary grid spacing which divides the drawing into cells. In other words, you can set the resolution of the sort. Note that smaller cell size does *not* imply that a more efficient sort of the Gerber output file will be achieved. Choice of optimum cell size depends on the size of features in the drawing. Through experience you will be able to determine optimum sorting cell sizes for particular applications. A good place to start, for example, would be to choose a cell size which is roughly the size of an IC socket (approximately 1-inch by 1-inch). Hence, if the user units are mils (0.001 inches) and the units factor is 1, an appropriate choice would be

```
    SORT PROXIMITY 1000,1000;
```

If both types of sorting are specified, the major sort will be based on aperture. Thus, components using the same aperture will be plotted first, and proximity sorting will take place within that group of components. Aperture is the major sort key and proximity is the minor sort key.

SORT PROXIMITY OFF specifies that proximity sorting is not to be done.

- > **SORT DRILL X ;**
- > **SORT DRILL Y ;**
- > **SORT DRILL OFF ;**

These commands allow you to specify whether or not the DRILL output coordinates are to be sorted. SORT DRILL OFF specifies to not sort the drill coordinates. SORT DRILL X sorts the drill coordinates by x-coordinate value (major sort on x-coordinate); if any x,y coordinates have the same x-coordinate value, the program sorts those coordinates by y-coordinate value (minor sort on y-coordinate). SORT DRILL Y specifies to sort the drill coordinates with a major sort on y-coordinate and a minor sort on x-coordinate.

Both SORT DRILL X and SORT DRILL Y order the tools from smallest to largest in the output file. However, neither command will change the tool numbers. Tool numbers assigned as drill tags (:T1030) appear in the input generate files (from top to bottom).

> **SOURCE file\_specifier [ file\_specifier ] ;**

> **SOURCE CLEAR ;**

This command allows you to specify the input generate files for PHOTOPLOT and DRILL. PHOTOPLOT and DRILL share the same set of input files. "File\_specifier" must be any legal Pascal file name(s), and this file must contain only generate information. If the "file\_specifier" does not end with \_g (the convention for HP EGS generate files), the suffix \_g will be automatically appended by PHOTOPLOT/DRILL. You may suppress this action by ending "file\_specifier" with a period. For example,

```
SOURCE myfile ;      means "myfile_g" will be used
SOURCE myfile. ;     means "myfile" will be used
SOURCE myfile_g ;    means "myfile_g" will be used
```

Many input files may be specified. In effect, PHOTOPLOT/DRILL will treat the entire set of input generate files as though they are one large generate file. SOURCE commands are cumulative in that

```
SOURCE myfile1 myfile2 ;      and
SOURCE myfile1 ;
SOURCE myfile2 ;
```

are equivalent. SOURCE CLEAR tells PHOTOPLOT/DRILL to "forget" previously entered SOURCE commands (this has the effect of "erasing the slate" and starting over).

> **DRILL UNITS FACTOR f ;**

> **PHOTOPLOT UNITS FACTOR f ;**

These commands allow you to specify a units conversion constant between the user units (used in the HP EGS drawing) and the units to be used for DRILL output and PHOTOPLOT output, respectively. For example, assume you are using mils (.001 inches) in the HP EGS drawing but wish to produce Gerber photoplotter code in units of inches. The proper command would be

```
PHOTOPLOT UNITS FACTOR 0.001 ;
```

since there are 0.001 inches in 1 mil. As another example, assume that you are using mils in the HP EGS drawing but wish to produce N/C drill code in units of inches. Then the proper command would be

```
DRILL UNITS FACTOR 0.001 ;
```

The internal default for both unit factors is 1. These commands can also be used to convert between metric and English units. You need simply use the proper factor, f.

**> VERSION ;**

This command identifies the version number of the PHOTOPLOT/DRILL post-processors program you are using.

**> VIEW ;**

This command displays the current status of all program parameters as well as all currently specified input generate files and PHOTOPLOT and DRILL output files. For example, a typical result from executing a VIEW command during a PHOTOPLOT/DRILL work session looks like this on the display screen:

```

AUTOFLASH ON:          ON
ERROR MODE:           CONTINUE ON ERROR
FILL WIDTH MODE:      OFF
SORT DRILL:           MAJOR SORT ON X
SORT PROXIMITY:       X: 1.00000E+001      Y: 1.00000E+001
SORT APERTURE:        ON

DRILL FORMAT:         4,2
PHOTOPLOT FORMAT:     2,3
DR ORIGIN SOURCE:     0.000E+000,0.000E+000  OUTPUT: 1.000E+002,1.000E+002
PH ORIGIN SOURCE:     -6.49E+003,-5.63E+003  OUTPUT: 1.000E+003,1.000E+003
DRILL UNITS FACTOR:   1.000E+000
PHOTOPLOT UNITS FACTOR: 1.000E+000
PHOTOPLOT RESOLUTION: OFF

NO FILL USE:          none specified
NOTE USE:             T15
POLYGON FILL USE:     T50

ATTRIBUTES FILE:     #5:/HP_EGS/NC/TESTS/PPtest_attr
WHEEL FILE:          #5:/HP_EGS/NC/fsd_wheel

```

Press <space bar> to continue OR <shift-execute> to stop.

CURRENT HP EGS GENERATE SOURCE FILES:

```
#5:/HP_EGS/NC/TESTS/PPtest_d
```

Press <space bar> to continue OR <shift-execute> to stop.

VIRTUAL DRILL FILE:

```
#5:/HP_EGS/NC/TESTS/PPtest_vdr
```

```
1 2 3 4 5 6 7 8 9 10 100
```

Press <space bar> to continue OR <shift-execute> to stop.



ACTUAL DRILL FILE:

#5:/HP\_EGS/NC/TESTS/PPtest\_adr

1 2 3 4 5 6 7 8 9 10 100

Press <space bar> to continue OR <shift-execute> to stop.

VIRTUAL PHOTOPLOT FILE:

#5:/HP\_EGS/NC/TESTS/PPtest\_vph

1 2 3 4 5 6 7 8 9 10

Press <space bar> to continue OR <shift-execute> to stop.

ACTUAL PHOTOPLOT FILE:

#5:/HP\_EGS/NC/TESTS/PPtest\_aph

1 2 3 4 5 6 7 8 9 10

>

Notice that a prompt character appears when all output files have been displayed.

- > **VIRTUAL DRILL file\_specifier layer\_list ;**
- > **VIRTUAL PHOTOPLOT file\_specifier layer\_list ;**
- > **VIRTUAL DRILL CLEAR ;**
- > **VIRTUAL PHOTOPLOT CLEAR ;**

These commands specify virtual type outputs for DRILL and PHOTOPLOT post-processors, and are otherwise identical to the ACTUAL DRILL and ACTUAL PHOTOPLOT commands. If you have trouble specifying an output file for a local disc, you are probably out of space. Please refer to the section on *Software Problems* in this chapter.

- > **QUIT ;**

This command terminates the current PHOTOPLOT/DRILL work session and returns you to the HP EGS main Manager menu or the main Pascal command line, depending on how you accessed the program.

> **WHEEL file\_specifier ;**

> **WHEEL CLEAR ;**

This command specifies the text file containing the aperture wheel information for PHOTOPLOT. "File\_specifier" is any legal Pascal file name. An error will result if the system cannot find the file. Examples of the WHEEL command are:

```
WHEEL #16:wheel1 ;
WHEEL #5:/MYDIR/wheel ;
```

Only one wheel file may be specified. If the WHEEL command occurs more than once, the most recent one will be used. The WHEEL CLEAR command specifies that the existing wheel file is to be cleared.

## Customizing the Message File

PHOTOPLOT/DRILL uses a special data file called a message file to provide internal data for operation. You may change some of this data for your application. This section describes how to change this message file.

The format of each entry in a message file is:

```
tag ^D^ information
```

You may change the messages in the information field, but you should not change the message tags or the delimiter (^D^). There is one exception, however, explained in the discussion of the CMDxxx message.

Although messages may be changed, they should *never* be entirely removed from the message file. If any messages contain the special character strings {#1} and {#2}, they will be replaced by runtime parameters when a message is accessed by the program.

The following entries in the message file may be changed. Be careful to change only these items described here, or unexpected errors may result.

### CODEWORDS ^D^

These are the security codewords needed to execute EWOPT:phdr.CODE. You must install a codeword before you can use PHOTOPLOT/DRILL.

**EXEC\_MEMORY\_FACTOR ^D^**

This entry is an integer constant which sets the average number of bytes of memory necessary to process one HP EGS component. Making this number larger reduces the likelihood of an out-of-memory error. Making this number smaller increases the size of the data blocks that PHOTOPLOT/DRILL can process. In general, this number should be made as small as possible but large enough to prevent either an out-of-memory error or memory thrashing (characterized by excessive processing time). If the system gives an out-of-memory error, the EXEC\_MEMORY\_FACTOR is too small and should be increased.

When you run the program, the system will place alpha characters (a, c, d, p) on the display screen for every component being processed. If the time between “c”s printed on the screen is unusually long (e.g., more than 2 seconds), the system is thrashing. This means the EXEC\_MEMORY\_FACTOR is too small and should be increased. Use care in adjusting its size, as this constant is set to a useful value when you receive PHOTOPLOT/DRILL.

**EXEC\_DEF\_FILE ^D^**

This is the default file's path name, which is input at startup time and after resets. This path name may be changed to point to any default file you wish to specify. In addition, you may specify a Yes or No option for the INPUT command. For example,

```
.,./EWDPT/Phdr_def N
```

is a legal string for this constant. This would mean that the default file is to be input with the No option (in response to Yes/No questions).

**EXEC\_FONT\_TABLE ^D^**

This allows you to choose the note stroke table you want to use to draw notes and text. The default file EWSYS:note\_st.TEXT is provided for you. The other note and text stroke tables provided for the HP EGS personalities are not useable with these post-processors. See Chapter 2: *Customizing Your System* for more information on creating new note stroke tables.

**PHxxxx ^D^**

These are Gerber output literals used by PHOTOPLOT. These constant literals may be changed to any desired character strings.

**DRxxxx ^D^**

These are Gerber output literals used by DRILL. These constant literals may be changed to any desired character strings.

**HELP\_xxx ^D^**

These are HELP command messages accessed by the system to provide on-line help. It may be desirable to provide specific help messages for your application. For example, you could append the following to the HELP\_DRILL message:

```
HELP_DRILL          ^D^
:
:
:
The convention is that all unplated holes are on
layers 122 and 123.

Use  ACTUAL DRILL #20:unplated 122 123
```

**CMDxxx ^D^**

**EXEC\_YES\_CHAR ^D^**

**EXEC\_NO\_CHAR ^D^**

Use care when changing the messages for

```
CMDxxx              ^D^
EXEC_YES_CHAR       ^D^
EXEC_NO_CHAR        ^D^
```

since these define literals that the program expects from you. CMDxxx represents command names the program expects from you. EXEC\_YES\_CHAR and EXEC\_NO\_CHAR are literal strings containing all characters recognized as Yes or No responses (answering (Y/N) questions). All CMDxxx messages should be unique and not null. Both EXEC\_YES\_CHAR and EXEC\_NO\_CHAR should be unique and not null. These messages should be changed with caution since command names may literally be changed. This could cause much confusion for others using the program if these command names are changed carelessly. Note that if any CMDxxx is changed, the corresponding HELP\_xxx will also have to change. These two literals must match indentically, or the HELP command will not be able to find the appropriate help message.

## Creating Library Parts

As everyone's wheel file is different, you must create your own library parts in order to run PHOTOPLOT/DRILL. This section outlines the procedure for creating library parts. In particular, this section describes how to use special tags :T1020, :T1021, :T1022, :T1024, and :T1030 to take advantage of features offered by PHOTOPLOT/DRILL.

### **:T1020 Flash Apertures**

By associating the special tag :T1020 with a circle or rectangle component, you may tell the program to flash a particular component with a particular flash aperture in the wheel file. The flash aperture name to be used is simply the value of the associated text string. For example, if you wish to use a particular oval-shaped aperture in a pad stack library part, you can do the following. Give the oval aperture in the wheel file a particular name, like "OVAL20", which means a 20-mil-wide oval aperture. This aperture would probably be a rectangular flash aperture of width 20 and length greater than 20. Next, associate the text string "OVAL20" and the tag value :T1020 with a circle or rectangle in the library part being created. The position of this circle or rectangle represents the position of the flash within the library part. The center of the flash aperture will be the center of circle or rectangle component. Now, when PHOTOPLOT/DRILL encounters this component, it will see the special associated text tag :T1020 and the name "OVAL20" and use this aperture to flash the given rectangle or circle component.

Rectangular flash apertures have a special property that circular and special flash apertures do not have. When PHOTOPLOT/DRILL encounters a flash aperture (by recognizing tag :T1020), it attempts to find this aperture in the wheel file. If not found it flags an error. If found it next checks its type to verify that it is a flash aperture. In the case of a rectangular flash aperture, one additional property is checked. If the component being flashed is a rectangle, the orientation (horizontal or vertical) of that rectangle must match that of the specified flash aperture. If the orientation is the same, the correct flash aperture is used. Otherwise, PHOTOPLOT will continue to search the wheel file for another flash aperture of that name. The search for a flash aperture name will continue until one of three conditions is met: a rectangular flash aperture with matching orientation, a non-rectangular flash aperture (circle or square), or a special aperture.

This feature allows you to specify two rectangular flash apertures with the same name, one having a vertical orientation and the other having a horizontal orientation. Thus, you can create only one library part for either horizontal or vertical orientation, instead of one for each orientation. When the library part is rotated by 90 degrees (from one orientation to the other), PHOTOPLOT will simply search the wheel file until it finds the flash aperture name with the appropriate orientation for the rectangle being flashed (horizontal or vertical). PHOTOPLOT recognizes rectangle apertures by orientation by observing that they are labeled as RECTANGLE\_H or RECTANGLE\_V in the wheel file data contained in the attributes file.

A wheel file may contain any number of apertures, and aperture names may be duplicated any number of times. While searching for an aperture name in the wheel file, PHOTOPLOT will act on the first matching name found (always scanning from the top of the file to the bottom). Thus, you may combine different physical aperture wheels in one wheel file. However, you must also assure that sets of different aperture names are disjoint, or you may get unexpected results.

### **:T1021 Polygon, Hatch, and Rectangle Fill Aperture**

You may want to use this aperture specification in a library part which contains components too large to flash (i.e., large rectangles) or irregular in shape (polygons and hatches). This tag value allows you to specify a particular trace aperture in the wheel file for filling a polygon, hatch, or rectangle. Use of this tag specifies that the component is to be filled by overlapping traces. In the case of a rectangle component, flashing takes precedence over filling with traces. Associating a text string with tag :T1021 specifies that a polygon, hatch, or rectangle is to be filled using the aperture named by the associated text string.

It may be convenient to develop macros that create rectangles, polygons, and hatches which are more efficient to plot. For example, a macro to draw large rectangles might work in the following way:

- draw a rectangle with filled width on the outside,
- draw a smaller filled rectangle on the inside, and
- associate the text string :T1021 to the outer rectangle, which specifies a smaller trace aperture than the default trace aperture.

Such a macro would allow you to specify a fairly large aperture for POLYGON FILL USE, providing for quick filling of large areas. However, it would specify a finer trace aperture for drawing the edges of the rectangle, providing sharp corners. The following shows this macro and its resulting figure.

```

DEF LRECT
{
  "Enter corner pts of large rectangle" X1 ;
  Y1
}
" ADD LR_RLAYER LR_LAYR {X1},{Y1} LRECT2 {X1} {Y1} " ;
DEF LRECT2
{
  X1 ;
  Y1 ;
  X2 ;
  Y2
}
" {X2},{Y2}; "
```

```

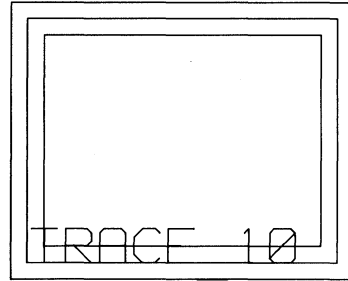
'DEF LR_OLD SNAP (" "+SYSTEM_SNAP_MODE+""); :RAW '
  "DEL LR_RLAYER LR_LAYR {X1},{Y1}; LRECT3 {X1} {Y1} {X2} {Y2} ";
  DEF LRECT3
  {
    X1 ;
    Y1 ;
    X2 ;
    Y2
  }
  "DEF LR_NEWX1 '(IFELSE({X1}>{X2},{X1}-(LR_WIDTH/2},{X1}+(LR_WIDTH/2)))'; "
  "DEF LR_NEWX2 '(IFELSE({X2}>{X1},{X2}-(LR_WIDTH/2},{X2}+(LR_WIDTH/2)))'; "
  "DEF LR_NEWY1 '(IFELSE({Y1}>{Y2},{Y1}-(LR_WIDTH/2},{Y1}+(LR_WIDTH/2)))'; "
  "DEF LR_NEWY2 '(IFELSE({Y2}>{Y1},{Y2}-(LR_WIDTH/2},{Y2}+(LR_WIDTH/2)))'; "
  "ADD LR_RLAYER LR_LAYR LR_WIDTH_SW LR_WIDTH :FILL "
  "LR_NEWX1,LR_NEWY1 LR_NEWX2,LR_NEWY2; "
  "ADD LR_NLAYER LR_LAYR :AC LR_NEWX1 LR_NEWY1 :T1021 LR_APER "
  "LR_NEWX1 LR_NEWY1 ; "
  "ADD LR_RLAYER LR_LAYR :FILL LR_NEWX1 LR_NEWY1 LR_NEWX2 LR_NEWY2; "
  " LR_OLD SNAP ;";
  DEF LR_NLAYER
  {
    X
  }
  "N{X}";
  DEF LR_RLAYER
  {
    X
  }
  "R{X}";
  DEF LR_WIDTH_SW
  {
    X
  }
  "W{X}";
  DEF LR_APER
  {
  }
  "TRACE_10";
  DEF LR_LAYR
  {
  }
  2;
  DEF LR_WIDTH
  {
  }
  100;

```

LRECT 0,0 1000,800 before turning on  
:AC text and turning off :FILL



LRECT 0,0 1000,800 after turning on  
:AC text and turning off :FILL. Note  
that the outer rectangle is tagged to  
be filled by the trace aperture  
TRACE\_10.



### **:T1022 Text/Note Draw Aperture**

You may specify a particular trace aperture in the wheel file for drawing notes and text. This may be useful when creating library parts which contain string designators such as pin numbers, part names, and board identification. By creating library parts which use specific apertures for different types of text, you can standardize different types of labels on a printed circuit board. These apertures may be specified only for text and note components. Associating a text string with tag :T1022 specifies that a text or note component is to be drawn using the aperture named by the associated text string.

### **:T1024 Ignore and No Fill Tags**

By associating the text string with tag :T1024 to a component, you may specify either an IGNORE or NO\_FILL operation for that component. You may not want some components, such as visual aids used to help draw a printed circuit board, plotted by PHOTOPLOT. To suppress plotting, associate the text string "IGNORE" with tag :T1024 to the selected component(s). PHOTOPLOT will simply ignore each component (as if it did not exist) when it encounters it in the input generate file.



In addition, you may not want some components filled by PHOTOPLOT, even though :FILL is specified for components or the FILL WIDTH mode is ON for a given run of PHOTOPLOT. To suppress filling, associate the text string "NO\_FILL" with tag :T1024 to the selected component(s). PHOTOPLOT will simply ignore the fill directive for each component (as if it did not exist) when it encounters it in the input generate file.

### **:T1030 Drill Holes**

When creating parts such as pad stacks, you will probably want to designate drill holes for an N/C drill machine. Drill holes may be specified by associating text with tag :T1030 to circles representing the holes in the HP EGS drawing. The value of the associated text string indicates the diameter of the tool to be used to drill the hole. If this value is null (blank) or an asterisk \*, the system assumes the tool size is the diameter of the tagged circle representing the hole. At run time, DRILL scans the input looking for drill holes (circles tagged with :T1030 associated text). When one is found, it prepares a drill hole at the output with a reference to the proper tool to be used. If it is the first hole of its size, DRILL creates a new tool reference. Once PHOTOPLOT/DRILL is complete, you may find a list of all tool references in the attributes file.

### **Other Considerations**

When creating library parts, remember to use no more than one special tag per component or else an error will result when PHOTOPLOT/DRILL is run. Also remember never to associate text to a drawing (by using the :AD option for associated text); instead always associate text to components (by using the :AC option for associated text) if you want to have this associated text recognized by PHOTOPLOT/DRILL. This is necessary because :AD associated text is disassociated from the drawing when the GENERATE command is used (:AD text is associated to instances, and the GENERATE command smashes all instances).

Only one level of associated text is recognized by PHOTOPLOT/DRILL. For example, associated text which is associated to another associated text component (i.e., two levels of association) is not recognized.

This program does not recognize implicit step information in a generate file. If this feature is never used in creating a drawing, it will not appear in a generate file (whether or not you use the GENERATE or GENERATE :I commands to create the file). However, if this feature is used, you must use the GENERATE :I command to smash the implicit steps into explicit steps.

An example of a typical library part using flashes and drill holes is shown in the HP EGS drawing EWEE:dipex. This is an example of an 8-pin dip similar to EWEE:dip8 except that it contains flash and drill hole designators (:T1020 and :T1030 associated text). This part contains two pad library parts EWEE:padexr and EWEE:padexo which are rectangular and oval pads; they are similar to EWEE:lp40r and EWEE:lp40o except that they contain flash designators. The associated text for flash designators (:T1020 associated text) is on layer 55. The associated text for drill hole designators (:T1030 associated text) is on layer 56. If you call up “EWEE:padexr” or “EWEE:padexo” from the Graphics Editor, you can see the flash and drill hole designators with the WIN :A; command. You can highlight the components associated to these designators by using the IDENTIFY command and selecting individual note components. The flash designators assume a wheel file containing the following aperture entries:

```
PADEXO      21 rectangle flash 80 55;
PADEXO      22 rectangle flash 55 80;
PADEXR      23 rectangle flash 75 55;
PADEXR      24 rectangle flash 55 75;
```

Here, PADEXO is an oval-shaped flash aperture and PADEXR is a rectangular-shaped aperture for flashing the different shaped pads. Notice that each name appears twice. This wheel file contains two different versions of each flash aperture: horizontal and vertical. Thus, if EWEE:dipex is rotated 90 degrees, PHOTOPLOT will automatically select the correct orientation.

Any library part may be changed to include flash designators. These flash designators will depend on the aperture wheel and aperture wheel file being used. Because of this, flash designator information does not appear in any of the standard library parts for the HP EGS Printed Circuit Board Layout personality. However, the pad parts (i.e., lp25s, lp46c, etc.) do contain drill hole designators (:T1030) with proper drill hole sizes.

## Software Problems

You may encounter some unexpected errors while running PHOTOPLOT/DRILL. If you do, check the following items and try to correct the error on your system. If you are unable to correct the problem, please keep a copy of your error trace file and contact your HP Sales Representative.

### Fatal Errors

Some fatal errors are not entirely unexpected. These errors are usually user mistakes, file I/O errors, or out-of-memory errors.

With a user mistake, the system will identify the problem for you. Often the solution is self-evident. For example, if you forget to specify a wheel file, the system will indicate this when a BEGIN command is executed. To correct the problem, you would specify a wheel file and try again.

Another fatal error is a file I/O error. For example, a user-specified output file may not be large enough to hold an entire set of output data. If this happens, the system will inform you and the current run of PHOTOPLOT/DRILL will terminate. The program will then give the prompt:

```
>
```

If you encounter an out-of-memory error, you will need to increase the EXEC\_MEMORY\_FACTOR specified in the message file EWOPT:phdr.ms.ASC.

### Output Files

You may specify the size of an output file by including [size] at the end of the file name. For example, if you want to create an output file 100 blocks in size, you could use

```
> ACTUAL PHOTOPLOT #20:Ph_output[100] 1 2 3 4 5
```

This would specify an ACTUAL PHOTOPLOT output file 100 blocks large called Ph\_output on logical volume #20. If a [size] is not specified, the Pascal file system will open an output file of maximum size. This has one problem, since you may be unable to specify additional output files on the same volume. This may occur because a volume is totally occupied by a previously specified output file. As a convenience you may specify a size of [\*]. This instructs the system to create an output file of half the maximum possible size on a given volume. In general, you may want to set aside entire (empty) volumes if large output files are expected. This will minimize the chances of running out of space part way through a run of PHOTOPLOT/DRILL.

**Units**

It is important that your HP EGS drawing units, aperture wheel units, and Gerber photoplotter units match whenever you use PHOTOPLOT/DRILL. If they do not, you may encounter a message telling you that the system cannot find the right size apertures in the wheel file for your drawing components. Check the following units statements for matching units (mils, inches, etc.): the UNITS command specified in your HP EGS personality's process file, the DRILL UNITS FACTOR and PHOTOPLOT UNITS FACTOR specified in your PHOTOPLOT/DRILL commands, and the units specified in the Gerber photoplotter code.

**File and Volume Names**

At times the system may not be able to locate files for a give PHOTOPLOT/DRILL run. Make sure the path names for the message file (entered when the program is started up), and the default file and the note stroke table (both located in the message file) are correct. If you have modified any of these files, check for syntax and format errors inside the files.

**PHOTOPLOT Format**

Make sure the numbers you specify in the PHOTOPLOT FORMAT command are appropriate for the resulting Gerber draw code. If they are not, you may encounter an out-of-range error when you try to use the PHOTOPLOT output file to run the photoplotter.

**Error Trace File**

The only errors reported in the error trace file are those indicating problems with the PHOTOPLOT/DRILL software itself. These errors should be uncommon. Once you start to receive messages in this file, the system will keep printing all the internal messages it finds until you leave the program with the QUIT command. Please be sure to include this listing when reporting software problems to your HP Sales Representative.

**Generate File Syntax Errors**

Syntax errors can occur while generate files are being read by PHOTOPLOT/DRILL. These errors should never happen if original HP EGS generate files are used. However, it is always possible that the data in a file is corrupted, or that a generate file is altered to an illegal format by another application program before the file is passed to PHOTOPLOT/DRILL. If a syntax error should occur, PHOTOPLOT/DRILL will inform you of the exact location (i.e., file name, line number, and token) where the error occurred. It will also supply you with an error number so that you may understand the nature of the problem. The current run of PHOTOPLOT/DRILL will terminate and the program will give the prompt:

>

The generate file syntax errors are:

```

11 = semicolon expected
12 = out of tokens
13 = command keyword token expected
14 = 'ADD' expected
15 = BEGIN/END or ADD expected
16 = illegal instance name
17 = illegal layer for this component
18 = unable to return token to token heap
19 = unable to fetch a legal token
20 = FILES keyword expected
21 = drawing name too long ($files$)
22 = #E expected
23 = lock angle not legal real valued
24 = lock angle out of bounds
25 = level designator out of bounds
26 = level designator not legal integer
27 = grid user units not legal real valued
28 = grid repeat factor not legal integer
29 = x-coor not legal real valued
30 = y-coor not legal real valued
31 = arc width not valid real value
32 = :R,:W,:FILL or x-coor expected
33 = circle resolution not legal real valued
34 = arc resolution not legal real valued
35 = circle width not legal real valued
36 = :R, :W, :FILL or x-coor expected
37 = :NV expected
38 = dimension scale factor not legal real valued
39 = dim switch :A,LA,LV,LH,D or :R expected
40 = illegal hatch specifier
41 = illegal hatch rot angle
42 = illegal hatch width
43 = not enough unhatch pts
44 = not enough hatch pts
45 = illegal instance scale factor
46 = illegal instance rot angle
47 = illegal line width
48 = illegal line scale factor
49 = not enough line pts
50 = illegal polygon width
51 = not enough polygon pts
52 = illegal rectangle width switch
53 = illegal text/note font switch
54 = illegal text/note rotation switch
55 = only text/note allowed within ASSOC block

```

56 = illegal text/note bind layer  
57 = no text/note bind layers  
58 = illegal literal  
59 = literal too long  
60 = illegal tag switch  
61 = illegal implicit step cols value  
62 = illegal implicit step rows value  
63 = illegal implicit step rotation value  
64 = illegal units type  
65 = illegal units number  
66 = illegal decimals number  
67 = illegal font integer  
68 = illegal resolution  
69 = illegal dim font  
70 = illegal dim decimal  
71 = illegal dim scale  
72 = illegal dim radix  
73 = illegal dim extend  
74 = illegal dim direction  
75 = illegal iname tag  
76 = illegal iname font  
77 = illegal iname layer  
78 = illegal iname parameter  
79 = illegal tname tag  
80 = illegal tname font  
81 = illegal tname layer  
82 = illegal tname parameter  
83 = equate label too long  
84 = :D, :S or :I expected in equate  
85 = illegal :L switch in equate  
86 = illegal :C switch in equate  
87 = illegal :P switch in equate  
88 = [ expected in equate  
89 = illegal bound layer in equate  
90 = illegal equate layer  
91 = illegal oval width  
92 = illegal oval resolution  
93 = illegal oval major radius  
94 = illegal oval minor radius  
95 = illegal oval start angle  
96 = illegal oval stop angle  
97 = illegal oval orientation angle  
98 = illegal slant in note/text  
99 = illegal fractional resolution for dim\_units  
100 = illegal dim\_units units specifier  
101 = illegal dim\_tolerance text size  
102 = illegal dim text slant  
103 = illegal text slant

# IGES Translator

## Chapter 6: IGES Translator

Introduction .....	229
Installing the IGES Translator .....	231
Local Installation .....	231
SRM Installation .....	231
Installing Your Codeword .....	232
Obtaining Your Codeword .....	234
The Codeword Installation Program .....	236
IGES File Format .....	237
HP EGS - IGES Relationships .....	238
The EGS-to-IGES Translator .....	239
Archiving Drawings .....	239
EGS-to-IGES Translator Default File .....	240
Using the EGS-to-IGES Translator .....	244
EGS-to-IGES Translator Messages .....	247
Limitations of the EGS-to-IGES Translator .....	248
Estimating Mass Storage Requirements and Run Time .....	249
The IGES-to-EGS Translator .....	250
Retrieving Drawings .....	250
IGES-to-EGS Translator Default File .....	251
Using the IGES-to-EGS Translator .....	253
IGES-to-EGS Translator Messages .....	254
Limitations of the IGES-to-EGS Translator .....	256
IGES-to-EGS Translator Process File .....	257
Estimating Mass Storage Requirements and Run Time .....	257





# IGES Translator

Chapter

6

## Introduction

This chapter describes the theory and operation of Hewlett-Packard's IGES 1.0 (Initial Graphics Exchange Specification) translator Revision 1.1, which is an HP EGS software option. The translator consists of two programs: an EGS-to-IGES Translator that converts HP EGS 2.0/2.1 archive files into IGES files, and an IGES-to-EGS Translator that converts IGES files into HP EGS 2.0/2.1 archive files.

IGES 1.0 is a standardized format for exchanging drawings among different CAD/CAM systems. It is an American National Standards Institute (ANSI) specification. For a complete technical description of IGES 1.0, you may order *Digital Representation for Communication of Product Definition Data*, publication ANSI Y14.26M - 1981, from:

The American National Standards Institute  
345 East 47th Street  
New York, New York 10017

Before reading this chapter, you should be familiar with the General Drawing personality presented in the first lessons of *Learning HP EGS*. You should also have a working knowledge of the Pascal Editor and Filer. If you need more information on using the Editor and Filer, please refer to *Managing and Editing Files with HP EGS*.

This chapter includes information on:

- Installing the IGES Translator and its codeword
- IGES file format
- HP EGS - IGES relationships
- Using the EGS-to-IGES Translator and its limitations
- Using the IGES-to-EGS Translator and its limitations
- Estimating mass storage requirements and run times

---

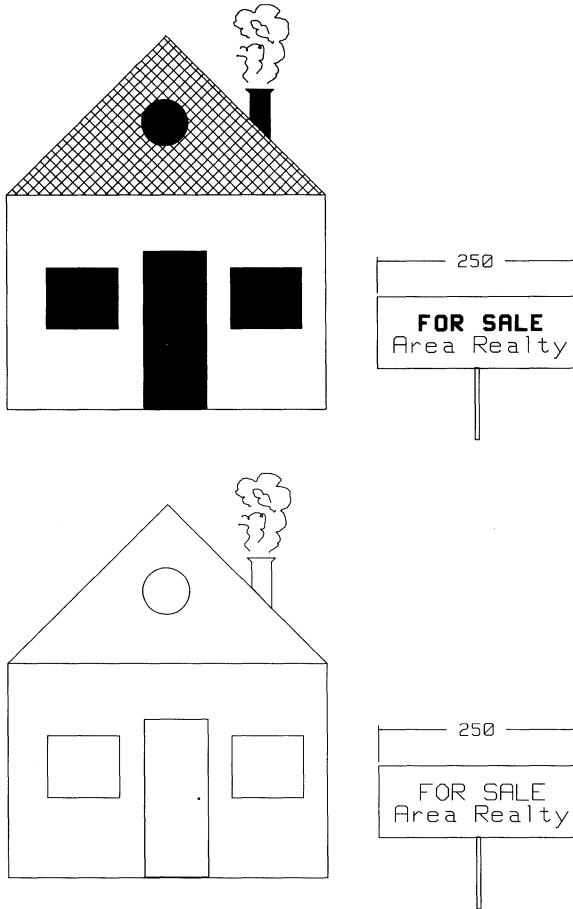
**Note**

Some HP EGS components are totally and irretrievably lost in translation, and some components being brought into HP EGS may not have equivalent components for translation. For a list of those components, see the sections on *Limitations of the EGS-to-IGES Translator* and *Limitations of the IGES-to-EGS Translator* later in this chapter.

To transfer drawings between HP EGS systems, use archive files and the appropriate storage and transfer media, not the IGES Translator.

---

The following examples show a simple HP EGS drawing of a house before translation into IGES (top) and after translation back into HP EGS (bottom) that will help you visualize the translation losses.



## Installing the IGES Translator

The following sections describe how to install the IGES Translator on either a local disc or a Shared Resource Manager (SRM). Read only the section that pertains to your system. These instructions assume you have previously installed the HP EGS software.

### Local Installation

1. Make sure that the Graphics Editor and the General Drawing Personality are installed on your system in the volumes EWSYS: and EWCODE:. These must be installed before you can use the IGES Translator. To verify you have installed these items, please refer to the verification procedure described in the *Installing HP EGS* manual.
2. Make sure that a volume named EWOPT: exists on your local disc. This is the volume where the IGES Translator will reside. If no such volume exists, return to the *Installing HP EGS* manual for instructions on creating a volume.
3. Insert the disc labeled IGES1 in unit number 3. This is the right drive in an HP 9836A/C and the left drive in an HP 9121D or HP 82901A Flexible Disc drive.
4. Starting from the HP EGS Manager menu, use the arrow keys to select System Utilities, and press **Return**.
5. Select the Pascal Command Line from this menu, and press **Return**.
6. Press **S** (for Stream) to initiate the installation procedure.
7. When the system prompts you for the name of the stream file, type in:

IGES1:iges\_lin **Return**

Then follow the instructions displayed on the screen. The system will ask you to insert each disc and press the space bar to continue with the installation procedure. The system will copy a total of 2 discs into the volume EWOPT:. When the procedure is finished, a message indicating successful completion will be displayed.

### SRM Installation

1. Make sure that the Graphics Editor and the General Drawing Personality are installed on your system in the directories EWSYS and EWCODE. These must be installed before you can use the IGES Translator. To verify you have installed these items, please refer to the verification procedure described in the *Installing HP EGS* manual.
2. Make sure that a directory named EWOPT exists on your SRM. If no such directory exists, return to *Installing HP EGS* for instructions first on creating an SRM directory.
3. Insert the disc labeled IGES1 in unit number 3. This is the right drive in an HP 9836A/C and the left drive in an HP 9121D or HP 82901A Flexible Disc drive.
4. Starting from the HP EGS Manager menu, use the arrow keys to select System Utilities, and press **Return**.
5. Select the Pascal Command Line from this menu, and press **Return**.

6. Press **S** (for Stream) to initiate the installation procedure.
7. When the system asks you for the name of the stream file, type in:

```
IGES1:iges_sin Return
```

Then follow the instructions displayed on the screen. The system will ask you for the name of the system directory and the optional product directory (e.g., #5:/HP\_EGS). Enter exactly the same information that was entered when you created the directory. The system will ask you to remove each flexible disc after everything is copied, insert the next disc, and press the space bar to continue with the installation procedure. The system will copy a total of 2 discs into the directory EWOPT. When the procedure is finished, a message indicating successful completion will be displayed.

## Installing Your Codeword

The IGES Translator requires a codeword in order to operate. Once you have installed the IGES Translator, you can follow the procedure in this section. One thing to note: If you live in the U.S. or Canada, the following procedure requires you to phone Hewlett-Packard during business hours (Mountain Time). If you live outside the U.S. or Canada, you must contact your HP Systems Engineer to obtain your codeword. It should take you about an hour to obtain and install your codeword.

A codeword is a 16-digit number that you enter once to permanently link your HP EGS software to a computer or HP-HIL ID Module. Codewords are also needed for the HP EGS personalities, the Connection Lister, and the optional PHOTOPLOT/DRILL post-processors. A unique codeword is needed for each workstation using the IGES Translator. A codeword is also needed for every workstation that operates the IGES Translator off of an SRM. A codeword is *not* a password that you type in each time you enter HP EGS.

In your HP EGS packaging you received a 7-by-9 inch envelope containing your IGES Translator codeword certificate. Do **NOT** open this envelope until you are certain you plan to keep your HP EGS IGES Translator software. Once the envelope has been opened, the system **cannot** be returned to HP.

The envelope contains a certificate similar to the one below:



### RIGHT-TO-USE CERTIFICATE FOR HP EGS (98305A)

CERTIFICATE SERIAL NUMBER

Please read this certificate. It tells you what you are entitled to with HP EGS and how to obtain your HP EGS codeword.

1. This certificate provides the serial number (printed above) and instructions you need to obtain your HP EGS codeword. You must have a codeword to save any HP EGS drawings. The codeword links your copy of HP EGS to a particular device (such as a computer or HP-HIL ID Module). As you follow the instructions for obtaining the codeword, remember that your copy of HP EGS can be run only on the device you specify.
2. This certificate entitles you, the purchaser, to use the one (1) copy of HP EGS (98305A) that is packaged with this certificate with one (1) computer or HP-HIL ID Module. You, the purchaser, specify the computer or HP-HIL ID Module when you obtain your HP EGS codeword.
  - This is a non-transferrable license to use 98305A according to Hewlett-Packard's rights and privileges statement. Under no circumstances is this Right-To-Use privilege transferrable.
  - 98305A is copyrighted and may not be copied (except for archive purposes, to replace a defective copy, or for program error verification).
3. To obtain your HP EGS codeword:
  - a. Select "System Utilities" from the HP EGS Manager menu. From the System Utilities menu select "Install Codeword".
 

The program finds the serial number and product number of each device that can be linked to your copy of HP EGS. For additional information on this step, see the section called "Installing Your HP EGS Codeword" in Installing HP EGS 2.1.
  - b. Write down the serial number and product number of the device that you plan to link to your copy of HP EGS.

MODEL \_\_\_\_\_ COMPUTER SERIAL NUMBER \_\_\_\_\_

- c. Obtain your codeword by:
  - Phoning (303) 226-3800 ext. 2222 if you live in the U.S. or Canada. Ask for the FEO Codeword Delivery Service.
  - Contacting your HP Systems Engineer or HP Application Support Representative if you live outside the U.S. or Canada.

The person you contact will ask for the certificate serial number at the top of this sheet and the two numbers you wrote down in the previous step.
- d. Write down the 16-digit codeword issued by the person you contact at  
 CODEWORD
- e. Hewlett-Packard.  
 To install your codeword, follow the steps in the section called "Installing Your HP EGS Codeword" of Installing HP EGS 2.1.
4. Retain this certificate for future reference. A Hewlett-Packard representative may need to see it at a later time.

PART NO. 98305-85001

## The HP EGS Right-To-Use Certificate

## Obtaining Your Codeword

1. Look at your certificate and note the serial number at the top.
2. If your computer displays the HP EGS Manager menu,

```

=>System Utilities
   General Drawing
   IGES Translator
  
```

skip to step 3. If not, continue.

- a. Turn off your computer (your computer only; do not turn off the disc).
- b. Turn your computer back on. **Hold down the space bar.**
- c. When you see 1E SYSTEM\_EGS at the top right corner of your screen, type in:

1E

The 1E will appear in the bottom right corner of your computer screen.

- d. The computer displays a screenful of HP EGS copyright information. Press the space bar to continue.
  - e. The computer prompts for the date. Enter the date and press **Return** or simply press **Return** to avoid the prompt.
  - f. The computer prompts for the time. Enter the time and press **Return** or simply press **Return** to avoid the prompt.
3. Select `System Utilities` from the HP EGS Manager Menu. To select a menu item, position the arrow by the menu item with the arrow keys. Press **Return**.
  4. Select `Install Codeword` from the System Utilities Menu. Press **Return**.

This utility runs a program to determine the product and serial numbers of your computer and/or any HP-HIL ID Module(s). You will see a display similar to the one below.

```
CPU ID Prom - Product number : 9836C  Serial number : 2250A00749
```

The CPU ID Prom is the identifier of your computer. In the example above, the product number is 9836C and the serial number is the 10-digit number 2250A00749.

If you have more than one module or both a computer and a module, you must decide which of these you want to link to your copy of IGES. After you obtain your codeword you can use your copy of IGES only with the computer or HP-HIL ID Module you have chosen. The advantage of linking your copy of IGES to a module is that you can take your copy of IGES and your module to other workstations.

5. Write down the product and serial numbers of the computer or HP-HIL ID Module that you plan to link with your copy of IGES:

Product Number: \_ \_ \_ \_ \_ Serial Number: \_ \_ \_ \_ \_

6. If you live in the U.S. or Canada, phone the number shown in step 3c of your codeword certificate. Ask for the HP EGS Codeword Delivery Service.

If you live outside the U.S. or Canada, contact your HP Systems Engineer or HP Application Support Representative.

7. The person you contact at Hewlett-Packard will ask for the:
  - Certificate serial number. This is listed at the top of your Codeword Certificate.
  - Product number of the device you are going to link to your copy of IGES. You wrote this down in step 5.
  - Serial number of the device you are going to link to your copy of IGES. You also wrote this down in step 5.
8. The person will then give you your 16-digit codeword. Write it both below and on your certificate.

CODEWORD: \_ \_ \_ \_ \_

Continue to the next section.

## The Codeword Installation Program

To install your HP EGS IGES Translator codeword, follow these steps. If you have any problems with the procedure, restart from step 1. To exit this utility, press **Stop**.

1. Continue from the Install Codeword utility explained in the previous section. Press the space bar until a prompt asks if you are ready to install the codeword.
2. The computer prompts you to enter a number between 1 and 4. These numbers represent the function for which you are installing the codeword. Enter:

4

to install the codeword for the IGES Translator, and press **Return**.

3. Type in your codeword as prompted. Then press **Return**.
4. The computer displays name of the message files (`toiges.ms,ASC` and `toegs.ms,ASC`) in which this utility will place your codeword. Simply press **Return**. Note: If you entered an SRM path name different than the one suggested in this manual, enter this path name in place of the one listed.

The system returns you to the System Utilities menu. To return to the HP EGS Manager menu, select `Return to Main Menu`.

The codeword should now be installed and you can use the IGES Translator. Note that you must install one codeword for each computer on an SRM system.



## IGES File Format

An IGES file is divided into five sections: Start, Global, Directory Entry, Parameter Data, and Terminate. The Start and Global sections are the only ones that contain user-modifiable information.

The IGES file itself is an ASCII (American Standard Code for Information Interchange) file. Only the comments in the Start Section are intended to be human-readable.

The five parts of the IGES file contain the following information:

### **Start Section**

The start section contains messages and comments added by the operator at run time. These comments appear in human-readable form at the beginning of the IGES file and serve as a memorandum to anyone who examines the file.

### **Global Section**

The global section contains file-handling information needed by the translator and some additional comments intended to help future users identify the contents of the IGES file. It also contains the input file name, system identification, product identification, scale factor, drafter's name, and organization name.

### **Directory Entry Section**

The directory entry section contains information on each entity of the drawing. The information is entirely determined by the translator and should not be modified. You should, however, be aware of the functions of two of the directory entry parameters. Parameter #5, the level number of the IGES file, corresponds to the layer number of the HP EGS file. Parameter #13, the pen number of a given level, is the same as the pen number assigned to the corresponding layer by the HP EGS process file.

### **Parameter Data Section**

The parameter data section contains data specific to each component of the translated drawing. These data are determined by the translator and need not be viewed or modified by the operator.

**Terminate Section**

The terminate section contains further file-handling information that is entirely determined by the translator.

## HP EGS - IGES Relationships

The following table indicates the relationship between HP EGS components and IGES 1.0 entities. For a thorough discussion of IGES entities, see *Digital Representation for Communication of Product Definition Data*, publication ANSI Y14.26M - 1981.

HP EGS Component (before translation) →	IGES Entity →	HP EGS Component (after translation)
Arc	Circular arc	Arc
Circle	Circular arc	Circle
Dimension :A	Angular dimension	Dimension :A
Dimension :LA, :LH, :LV	Linear dimension	Dimension :LA, :LH, :LV
Dimension :D	Diameter dimension	Dimension :D
Dimension :R	Radius dimension	Dimension :R
Hatch	NOT TRANSLATED	
Instance	Subfigure instance	Instance
Line	Copious data <sup>1</sup>	Line
Line :C	Copious data	Line
Line :L	Leader	Line :L
Marker	Point	Marker
Oval	Conic arc	Oval
Polygon	Copious data	Polygon
Rectangle	Copious data	Rectangle
Text	General note	Note
Note	General note	Note
	Line	Line
	Composite Curve	Individual Components
	General Label	Leader lines & note
	Section Entity	Lines
	Unsupported IGES Entity	NOT TRANSLATED

<sup>1</sup> "Copious data" is an IGES entity that refers to a series of points.

## The EGS-to-IGES Translator

The EGS-to-IGES Translator converts HP EGS archive files into IGES 1.0 files. The IGES files may then be sent to any IGES-equipped CAD/CAM systems. After translation, you will have both an IGES file and the original HP EGS archive file.

Keep in mind that an IGES file is not an actual drawing, but a file of data that will eventually be converted back into a drawing on some other system. The final appearance of a drawing is determined by the destination system and its own IGES translator.

This section describes:

- Archiving drawings for translation into IGES
- The EGS-to-IGES Default File
- Using the EGS-to-IGES Translator
- EGS-to-IGES translator messages
- Limitations of the EGS-to-IGES Translator
- Estimating mass storage requirements and run time

### Archiving Drawings

Only files archived in HP EGS 2.0 may be processed in the EGS-to-IGES Translator. Be sure the process file used to create the drawing is loaded into HP EGS 2.0 before archiving the drawing. If your drawings are stored as EGS/200 versions 1.0/1.05, just re-archive in 2.0. For EGS/45 archive files, you must retrieve and re-archive these drawings in HP EGS 2.0.

When archiving an HP EGS file for translation (with the ARCHIVE command), use the :P option which archives the process file along with the drawing itself, and the :M option which converts macro instances into ordinary instances. If your drawings contain implicitly stepped components, you must also use the :I option to change them into explicitly stepped components. For further details on using the ARCHIVE command, please refer to the *HP EGS Syntax Reference*.

If your drawing contains instances and the target system's translator cannot process subfigure definitions and instances, use the GENERATE command (rather than the ARCHIVE command) to produce a file that has all the library parts smashed.

## EGS-to-IGES Translator Default File

Before using the EGS-to-IGES translator, you must understand the parameters the program will ask you to supply. These parameters have default values which you may change. The default file for the EGS-to-IGES Translator (EWOPT:toiges\_df.ASC) contains the following information:

```

scale          ^D^1.00000
drafter        ^D^
organization    ^D^
start_section  ^D^

note_st        ^D^note_st,TEXT
text_st        ^D^text_st,TEXT
note_form      ^D^4
text_form      ^D^4
l_arrow_form   ^D^2
d_arrow_form   ^D^2

line_weight    ^D^Y
copious_form_11 ^D^Y

tape_select_addr ^D^1404
bpi            ^D^1600
block_factor   ^D^10

l_arrow_height ^D^0.042
l_arrow_width  ^D^0.125
d_arrow_height ^D^0.042
d_arrow_width  ^D^0.125

solid          ^D^1
dashed         ^D^2
long_dash      ^D^2
dot_center     ^D^4
dash_center    ^D^4
phantom        ^D^3
long_dotted    ^D^2
dotted         ^D^2

degree         ^D^179
plus_minus     ^D^160

```

The parameter names are in the left column and the defaults are in the right column. When you use the text editor to modify the defaults, work only on the items to the right of the delimiter ^D^. Everything to the left of ^D^ is a message tag and should not be changed.

Any new default values you type in this file will appear on the screen in place of the old defaults every time you run the translator.

### Memo Parameters

The first four items in the file are memo parameters. They have no effect on the drawing itself; they merely tell other users about the stored drawing. (IGES files are standardized and self-identifying so that they can stand alone without external documentation.)

In the default file shipped with the translator software, all the memo parameters except the scale factor have blank defaults. To make your work easier, you may edit the default file and replace them with defaults tailored to your application.

All but one of the memo parameters are limited to one line of text. The exception is start section comments, which may contain as many lines of text as desired.

The memo parameters are:

Scale factor	The scale factor of the drawing. Note: the scale factor must be a positive, real number, such as 0.125. Ratios, such as 1:8; fractions, such as 1/8; and text are not allowed. The scale factor has a default of 1.00000, indicating an actual-size drawing.
Drafter	The author(s) of the drawing.
Organization	The name of your company, department, etc.
Start section comments	Other comments appear in the translated IGES file for the drawing. Comments usually include information about the drawing that another user may need for future reference. This section appears blank until you enter your desired information. This section may be as long as you wish. This section will be put into the start section of the IGES file unless it is changed at run time.

### Form parameters

The next six user-modifiable parameters in the default file govern the appearance of the lettering and arrowheads in the IGES file. Appropriate defaults are provided in the default file shipped with the translator, but you may want to change them to meet some special need.

Note and text stroke files	The note and text stroke files contain the information the translator needs to draw characters. Default stroke tables are provided with HP EGS, but you may substitute your own if you wish. For further information, see Chapter 4: <i>Customizing your System</i> .
----------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Note and text form numbers HP EGS notes and text are both translated into IGES “general notes,” as IGES has no text entity. However, HP EGS text is given a line weight of 3 when translated, and HP EGS notes are given a line weight of 0. Because lines with weight in IGES are always solid lines, HP EGS text is translated into something resembling filled text.

There are six available IGES fonts. The fonts have different character sets as well as different appearances. Number 4, which best matches the standard HP EGS note and text, is the default font. If you wish, you can use one IGES font to represent text and another to represent notes.

**IGES Note Fonts<sup>1</sup>**

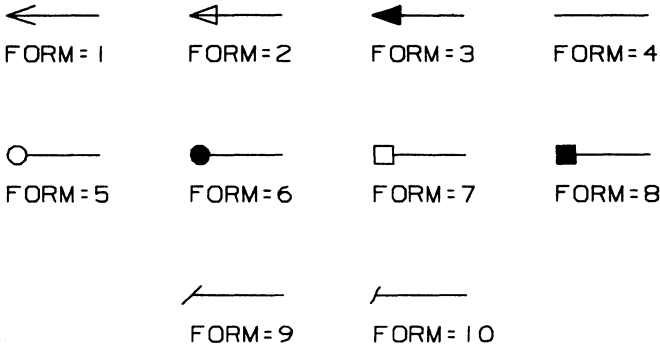
FONT CODE  
(FC)

FONT APPEARANCE

1	ABCDEFGHIJKLMN OPQRSTUVWXYZ 1234567890I "#%&'()x+,-./:;?
2	ABCDEFGHIJKLMN OPQRSTUVWXYZ1234567890%()*+,-./:;
3	ABCDEFGHIJKLMN OPQRSTUVWXYZ1234567890() +,-./:;
4	ABCDEFGHIJKLMN OPQRSTUVWXYZ 1234567890I "#\$%&'()*+,-./:;?
5	ABCDEFGHIJKLMN OPQRSTUVWXYZ 1234567890I "#\$%&'()*+,-./:;?
6	ABCDEFGHIJKLMN OPQRSTUVWXYZ 1234567890I "#\$%&'()*+,-./:;?

Leader arrow/dimension arrow forms The 10 different arrowheads shown below are available in IGES for use with leader and dimension lines. Choose the ones that best suit your needs. The default value is 2.

**IGES Arrowheads<sup>1</sup>**



<sup>1</sup> Font and arrowhead illustrations are from ANSI Y14.26M - 1981 courtesy of American Society of Mechanical Engineers. Used by permission.

### Other Default Options

Other options available from the default menu are:

- Translate width into line weight: You may translate component widths into IGES in two ways. If you choose Y for Yes, width becomes line weight in the directory entry for that entity. If you choose N for No, the component with width will be converted to a series of entities which represent its pictorial equivalent (i.e., lines will be converted to polygons; circles, rectangles, and polygons will be converted to two concentric circles, rectangles, or polygons; arcs and ovals will be converted to two arcs or ovals and two lines). The default value is Y.
- Output copious data form 11: You may translate an HP EGS line, polygon, or rectangle into copious data form 11 or 12. Form 11 is a two-dimensional linear string (or multi-vertex line), while form 12 is a three-dimensional linear string. If you select Y for Yes, the translator will output copious data form 11. If you select N for No, the system will output form 12. Choose the one that can be processed by your target system's translator.

### Tape Driver Parameters

The next three items are tape driver parameters:

- tape\_select\_addr The logical address of your tape drive (1404 in the case of the HP Model 7974 tape drive recommended for use with HP EGS).
- bpi Bits per inch. We recommend 1600 bpi if that is compatible with the destination system's tape drive.
- block\_factor We recommend a blocking factor of 10 if that is compatible with the destination system's tape drive.

### Default Dimension Arrowheads

These values are only used when a generate file is supplied as input to the EGS-to-IGES translator.

- L\_arrow\_height Default height for leader line arrowheads.
- L\_arrow\_width Default width for leader line arrowheads.
- d\_arrow\_height Default height for dimension arrowheads.
- d\_arrow\_width Default width for dimension arrowheads.

### Line Type Relationships

The next eight items in the default file determine how to translate HP EGS line types into IGES line types. The HP EGS line styles are in the left column and the IGES line form numbers are in the right column.

solid	Solid lines are translated into solid lines (IGES Type 1).
dashed	Dashed lines are translated into dashed lines (IGES Type 2).
long_dash	Long-dash lines are translated into dashed lines (IGES Type 2).
dot_center	Dot-center lines are translated into centerlines (IGES Type 4).
dash_center	Dash-center lines are translated into centerlines (IGES Type 4).
phantom	Phantom lines are converted into phantom lines (IGES Type 3).
long_dotted	Long-dotted lines are converted into dashed lines (IGES Type 2).
dotted	Dotted lines are translated into dashed lines (IGES Type 2).

### Symbols

The next two items specify how to translate symbols:

degree	ASCII character 179 (the degree symbol) is converted to \$ (IGES Form# 1002).
plus_minus	ASCII character 160 (the plus_minus sign) is converted to # (IGES Form# 1002).

### Using the EGS-to-IGES Translator

The following section describes a procedure for using the EGS-to-IGES Translator. Please note that all file names you type in will be converted to lowercase except for the suffixes .TEXT or .ASC. Volume names and SRM path names remain uppercase.

1. Use the arrow keys and **Return** to select the IGES Translator from the HP EGS main Manager menu.
2. Use the arrow keys to select EGS to IGES from the IGES Translator menu. A series of prompts will now appear on your CRT.
3. The first prompt asks you to enter the name of the HP EGS archive/generate file to be translated. Type in the complete file specification for your system, including the suffix .a or .g. No default suffix is assumed.
4. The second prompt asks you to enter the output destination. You have two options: writing to a file and writing to a magnetic tape.



---

**Note**

To run both a local disc and a magnetic tape drive with your unit, use a separate disc interface card (HP 98625A) for the magnetic tape drive.

---

**a. Writing to a File:**

Enter a complete file specification, or simply press **Return** to select the default output file name. The default is the input file name, with the last two characters removed to give the file a unique name.

**b. Writing to a Magnetic Tape:**

Enter the keyword `mastape`. The program then gives you four magnetic tape prompts.

The four magnetic tape prompts are:

Enter the mastape select code and address (default = 1404):

Enter the proper select code and address for your system. Or simply press **Return** without typing in anything to keep the default.

Tape density (default = 1600):

1600 bits per inch is recommended for IGES tapes. Enter a density (800 or 1600 bpi) appropriate for the destination system's tape drive, or press **Return** to keep the default.

Blocking factor (default = 10):

A blocking factor of 10 is recommended for IGES tapes. Enter a blocking factor appropriate for the destination system's tape drive, or press **Return** to keep the default. The maximum block size allowed is 204.

File number to write to (0 to append):

Enter the file number for your IGES output file. Enter 0 to begin writing at the end of the last file now on the tape.

5. The next prompt asks you to enter the log file destination (either your CRT, a file path, or the keyword `PRINTER:`). Pressing **Return** defaults to your CRT. Because the log file contains error and comment messages from the computer, you may wish to choose the printer or a file specification so you will have a permanent copy.

---

**Note**

Be sure to use different file specifications, including a different volume name, for the log file and the output file.

---

6. You will be asked whether you want to change the default parameters. If you answered Y to change the remaining defaults, the IGES-to-EGS Default Menu shown below will appear on the screen. Use the arrow keys to select the item you want to change, press **Return**, type in the replacement for the default, and press **Return** again.

```
-->  Scale factor: 1.00000
      Product name:
      Organization:
      Drafter:
      Start section comments:
      Note stroke file name: note_st,TEXT
      Note form number: 4
      Text stroke file name: text_st,TEXT
      Text form number: 4
      Leader arrow form: 2
      Dimension arrow form: 2
      Translate width into line weight: Y
      Output copious data form 11: Y
      RUN
```

Your replacement answers are limited in length to the width of the screen except for start section comments. Refer to the *EGS-to-IGES Translator Default File* section for the allowable values.

All the lines on the menu except product name have a corresponding value in the default file. Type in the name of the product depicted in the drawing for product name.

Selecting start section comments erases what was last entered in this section, but it *does not* delete the comments permanently stored in the default file. When you select this item, the screen is cleared except for the following reminder at the top:

```
Start section comments (<RETURN> <RETURN> to return to menu):
```

You may type in as much alphanumeric text as you wish. At the end of each line, press **Return** once. When you are done, press **Return** twice to return to the default menu.

To go back to the original default for any given item, use the arrow keys to select that item over again, and then press **f2** (the **RECALL** softkey) to the desired answer and then press **Return**.

When you are satisfied with the answers to all the items in the menu and are ready to run the translator, select RUN and press **Return**.

7. During the translation, you can press **Stop** to abort the process. If the translation is completed without any errors, you will see the following messages:

```
Translation was successful. Number of warnings:
Translation finished at (date) (time)
Do you have another file to translate? (Y/N)
```

Typing Y for Yes restarts the EGS-to-IGES Translator. Typing N for No returns you to the IGES Translator menu.

After a successful translation, you will have both an IGES file and the HP EGS archive file it was made from. If the translation is unsuccessful, you will see:

```
Translation aborted
```

along with the appropriate error messages. The system will then return you to the IGES Translator menu.

## EGS-to-IGES Translator Messages

The translator outputs warning messages for components not translated into IGES. Examples of some messages are:

- HP EGS hatch not translatable; hatch component deleted

HP EGS hatch components cannot be conveniently represented in IGES 1.0. Thus, the hatch component is not translated.

- Area fill not allowed in IGES; filled areas emptied

IGES 1.0 does not have any representation of area fill. Thus, area fill of closed areas is not translated.

- Implicitly stepped component not translated; re-archive with :I option

The IGES translator translates only the original component; the other components in the array are not translated into IGES. You must re-archive the drawing with the :I option to step the instances explicitly.

- Construction line does not intersect with window.

A construction line may lie outside the window, and thus will not be translated into the IGES file.

- IGES has no construction lines; line clipped to edge of window.

IGES does not have construction lines. The translator translates a construction line into a line segment if it lies within the boundary of the drawing.

- routine name:xxxx , error number:3

There is not enough current memory to translate the drawing.

## Limitations of the EGS-to-IGES Translator

The HP EGS components described below either cannot be translated into IGES or change in appearance when translated.

:AC and :AD text	Text associated with a component (added with the :AC option) is translated into IGES, but the logical location, and logical location layer are lost in translation. The tag value is translated into an entity label subscript in the directory entry for the entity. Associated text is tied to its component via the pointers to general note at the end of the parameter record. Text associated with a drawing (added with the :AD option) for subfigure definition is lost in translation.
Area fill	IGES 1.0 has no area fill feature for closed shapes. Filled HP EGS components are left unfilled.
Line type	<p>IGES provides four line types: solid (1), dashed (2), phantom (3), and centerline (4). HP EGS provides eight line types: solid, dashed, long dash, dot center, dash center, phantom, long dotted, and dotted. The translator maps line types not supported by IGES (long dash, dotted, long dotted, and dot center) to their closest representation.</p> <p>Dotted and long dotted lines in an HP EGS file are translated into dashed lines in IGES. Dot-center lines are translated into centerlines. Long-dash lines are translated into dashed lines. You can change those relationships if you wish by making the appropriate changes to the default file.</p>
Construction lines	Construction lines, which are infinitely long in HP EGS, cannot be represented as infinitely long in IGES. The EGS-to-IGES Translator clips them at the edges of the HP EGS drawing window. Any construction lines inside an instance or library part are clipped at the edges of the part.
Hatching	Hatching cannot be represented in IGES 1.0. All hatching in the HP EGS archive file is lost in translation, leaving blank areas where the hatching pattern had been.

Implicit stepping	Components that are implicitly stepped in the HP EGS drawing file must be explicitly stepped using the :I option when the file is archived. Otherwise only the original component will be retained; its duplicates will be lost in translation.
Text	Since IGES has no text entity, HP EGS text is translated into IGES "general notes" with a line weight of 3, giving the effect of filled text.
Macro instances	Macro instances are translated into ordinary instances if you used the :M option when archiving the HP EGS file. If you do not use the :M option, the translation will be aborted.
Line width	HP EGS line width is by default translated into IGES line weight. Lines with width in HP EGS will always appear to be filled in IGES, because IGES defines a line with weight as a solid line. If you choose to translate width into a series of components, the width property will be lost.

### Estimating Mass Storage Requirements and Run Time

The size of the IGES file produced by the translator and the time required for translation are directly related to the size of the HP EGS archive file. The IGES file is about five times the size of the HP EGS archive file, and translation takes about three seconds per kilobyte of memory occupied by the archive file.

If the size of the IGES file exceeds the available space on your mass storage medium, the translator responds with the prompt:

```
Current output file full. Do you want to write the rest of
the output to another file? (Y/N)
```

If you type in Y, you will see:

```
Output destination:
```

Enter a new file specification or mount a new tape and repeat the magnetic tape procedure.

If you type in N, the translation is aborted.

## The IGES-to-EGS Translator

The IGES-to-EGS Translator converts IGES 1.0 files into HP EGS archive files. After being retrieved from an archive file, the translated files may be treated like any other HP EGS drawing files. After translation, you will have both an HP EGS archive file and the IGES file from which it came.

Three-dimensional data that is given a proper transformation in IGES into two-dimensional views can be extracted by the IGES-to-EGS Translator. Each two-dimensional view will be extracted only when it has a corresponding “drawing entity” in the IGES file. If there are two or more drawing entities in the IGES file, the drawing names will be enumerated.

This section describes:

- Retrieving drawings into HP EGS
- The IGES-to-EGS Default File
- Using the IGES-to-EGS Translator
- IGES-to-EGS Translator messages
- Limitations of the IGES-to-EGS Translator
- Estimating mass storage requirements and run time

### Retrieving Drawings

The output file from the IGES-to-EGS Translator must be retrieved in the HP EGS Graphics Editor (using the RETRIEVE command) to create an HP EGS drawing. The procedure for retrieving the file is as follows:

1. Select an appropriate personality for your drawing from the HP EGS main Manager menu. Make sure to load the proper process file, note and text stroke files, and arrowhead drawings. These are normally loaded for you when you select a personality.
2. Make sure that there are no duplicate file names in the volume where you plan to store the new drawing. The RETRIEVE command will overwrite any files with the same name, and you could unintentionally lose some previously stored drawing information.
3. Type in:

```
RETRIEVE input file specification output volume specification ;
```

4. When the retrieve is done, you may view your entire drawing by typing in:

```
WIN :F ;
```

You may then manipulate the drawing like any other HP EGS drawing.

## IGES-to-EGS Translator Default File

Before using the IGES-to-EGS translator, you must understand the parameters the program will ask you to supply. These parameters have default values which you may change. The default file for the IGES-to-EGS Translator (EWOPT:toegs\_df.ASC) contains the following information:

```
tape_select_addr  ^D^1404
default_layer     ^D^255
close_poly       ^D^Y
preserve_layer    ^D^Y
extract_tag       ^D^N
window           ^D^Y
no_clipping_plane ^D^N
default_subfig    ^D^XYZZY

degree           ^D^179
plus_minus       ^D^160
```

The parameter names are in the left column and the defaults are in the right column. When you use the text editor to modify the defaults, work only on the items to the right of the delimiter ^D^. Everything to the left of ^D^ is a message tag and should not be changed.

The new defaults you type in will appear on the screen in place of the old defaults every time you run the translator.

### Default Values

The items in the IGES-to-EGS default file describe the following:

- |                  |                                                                                                                                                                                                                                                                                                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tape_select_addr | The logical address of your tape drive (1404 in the case of the HP Model 7974 tape drive recommended for use with HP EGS).                                                                                                                                                                                      |
| default_layer    | The default layer is the one that receives all IGES entities that are on levels not defined in HP EGS. Choose the layer (1 through 255) that best suits your needs.                                                                                                                                             |
| close_poly       | If the answer is Y for Yes, all closed lines in the IGES file will be converted into rectangles or polygons in HP EGS. If the answer is N for No, they will be left as closed lines.                                                                                                                            |
| preserve_layer   | If you choose No for the prompt about preserving layers, the translator will assign a layer number to entities which have the same original layer, line font number, and pen number. If the translator runs out of layers, the default layer will be used for all remaining layer/font/pen number combinations. |

If you choose Yes, and if there is more than one line type or pen number assigned to the layer, the most frequently used combination of line type and pen number will be chosen for the layer.

If the pen number 8 in HP EGS (black on the CRT) is used, it will be converted to pen number 1 so that the image will be visible on the CRT.

`extract_tag` If you choose Yes for the prompt about translating the entity label subscript, the number in the field will become the tag value for associated text. If you choose No, the number in the entity label subscript field will be ignored.

`window` If you choose Yes, the translator will output the WINDOW command in the archive file. This means the system will fit the drawing within this window before you retrieve it. This also means the translation will take longer.

If the window of the drawing is not important, you are advised to set this option to n for No.

`no_clipping_plane` If you choose Yes, clipping planes in the view entity will be ignored. This option must be set to Yes for ComputerVision IGES files.

`default_subfig` The name is used as a prefix to form a new name for the library part when an invalid name is found in subfigure definitions. The first such part will be named `xyzzy00`, the next `xyzzy01`, and so on.

`degree` \$ (IGES Form# 1002) is converted to ASCII character 179 (the degree symbol).

`plus_minus` # (IGES Form# 1002) is converted to ASCII character 160 (the plus-minus sign).



## Using the IGES-to-EGS Translator

The following procedure describes how to use the IGES-to-EGS Translator. Please note that all file names you type in will be converted to lowercase except for the suffixes .TEXT or .ASC.

1. Select the IGES Translator from the HP EGS main Manager menu.
2. Select the item named IGES to EGS from the IGES Translator menu. A series of prompts will now appear on your screen.
3. The first prompt asks you to enter the name of the IGES file to be translated. Use either a complete file specification for your system or the keyword `mastape`.

---

### Note

To run both a local disc and a magnetic tape with your computer, use a separate disc interface card (HP 98625A) for the magnetic tape unit.

The IGES-to-EGS translator expects the magnetic tape to be unformatted.

The IGES file must be ASCII format without any extraneous characters.

---

- a. If you enter a file specification, go to step 4.
- b. If you enter the keyword `mastape`, the program gives you two magnetic tape prompts.

Enter the `mastape` select code and address (default = 1404):

Type in the proper select code and address for your system, and press **Return**.  
Simply press **Return** without typing in anything to keep the default.

File number to translate:

Enter the number of the IGES file you want to translate, and press **Return**. There is no default for this prompt.

4. The next prompt asks you for the output file destination. Enter a complete file specification.

Be sure to avoid file name conflicts. If there is a file name conflict, you will see the message:

File already exists. Do you want to overwrite it? (Y/N):

If you type in Y, the program overwrites the existing file. You can then go on to Step 5. If you type in N, you will return to Step 4.

5. The next prompt asks you to enter the log file destination (either a file specification or the keyword `Printer:`). Pressing **Return** defaults to your computer's CRT. Because the log file contains error and comment messages from the computer, you may wish to choose the printer or a file specification so you will have a permanent copy.
6. You will be asked whether you want to change the default parameters. If you answered Y to change the remaining defaults, the IGES-to-EGS Default Menu shown below will appear on the screen. Use the arrow keys to select the item you want to change, press **Return**, type in the replacement for the default, and press **Return** again.

```
--> Translate closed lines into polygons? (Y/N): Y
    Default layer: 255
    Preserve layers (Y/N): Y
    Translate entity label subscript into tag value (Y/N): N
    Output window command: Y
    Ignore clipping planes in view entity: N
    RUN
```

When you are satisfied with the answers to all the items in the menu and are ready to run the translator, select RUN and press **Return**.

7. During the translation, pressing **Stop** aborts the process. If the translation is completed without any fatal errors, the following message will appear on your computer's CRT:

```
Translation finished at (date) (time)
Do you have another file to translate? (Y/N)
```

Typing Y for Yes restarts the EGS-to-IGES Translator. Typing N for No returns you to the IGES Translator menu.

If the translation is successful, you will have both an HP EGS archive file and the IGES file it was translated from. You may turn the archive file into an HP EGS drawing file using the RETRIEVE command.

If for any reason the translation is unsuccessful, you will see the message:

```
Translation aborted
```

along with the appropriate error messages. You will then return to the IGES Translator menu.

## IGES-to-EGS Translator Messages

Two kinds of messages are sent out by the IGES-to-EGS Translator. Error messages indicate problems with the input file, the hardware, or the translation process. Comment messages indicate major steps in normal operation.

These messages indicate problems inside the IGES file. If such a message appears, try obtaining another copy of the IGES file and checking the storage and transfer media used in the translation. Most of the errors described below simply cause the entity to be ignored; there are two fatal errors listed.

### Global Section Errors

- Invalid file name in Global Section.

The parameter in field #4 is not a valid HP EGS file name. You will be asked to enter a file name for the drawing.

### Directory Entry and Parameter Data Sections Errors

- Record numbers are out of sequence.

The number at the end of each record is not in the proper sequence (ascending order and increment of 1). This is a fatal error. The translation will be terminated.

- Unexpected end-of-file.

The IGES file is terminated abnormally (i.e., not by the terminate section). This is a fatal error. The translation will be aborted.

- There is no entity number.

The entity number is missing in either the directory entry record or the parameter data record.

- The entity numbers do not match.

The entity number in the parameter data record does not match that in the corresponding directory entry record, or the pair of entity numbers for a particular entity in its directory records do not match.

- Directory entry pointer is missing.

The parameter data record does not have a valid directory entry number in column 65-72.

- Unsupported entity.

The entity is not supported by the IGES-to-EGS translator.

- Unexpected end-of-record.

The translator encounters end of record delimiter before all parameters for the entity are extracted from the parameter record.

- String lengths do not match.

The number of characters in a string does not match the number preceding the delimiter letter H at the beginning of the string.

- `Bad data format.`

The value found for the parameter does not conform with the expected data type (i.e. numeric value for string and vice versa).

- `Missing parameter data value.`

No default is allowed for the specific parameter for a specific entity.

- `routine name:xxxx , error number:3`

There is not enough current memory to translate the drawing.

## Limitations of the IGES-to-EGS Translator

IGES 1.0 is a broad-based standard that includes some engineering graphics entities not found in HP EGS. All IGES entities that have no HP EGS equivalents will be eliminated by the translator.

The IGES entities discussed in this section are not translated into HP EGS in exactly their original form.

Dimensions	In IGES, each dimension includes leader lines, witness lines, and a dimension note. The IGES-to-EGS Translator converts each dimension in the IGES file into the nearest HP EGS equivalent. The appearance may change slightly in the process, but the essential information is retained.
Vertical text	IGES vertical text (stacked individual letters) is translated into horizontal notes.
Note fonts	IGES general notes are translated into HP EGS notes. The appearance of the translated notes is determined by the note stroke file.
Arrowheads	The IGES-to-EGS Translator uses the library parts <code>l_arrow</code> and <code>d_arrow</code> to draw independent leader lines and leader lines associated with dimensions, respectively. These library parts are supplied with HP EGS and are automatically found by the translator. If the arrowheads used in the IGES file differ from the default arrowheads, create drawings of the appropriate arrowheads such that your pictures are drawn correctly. Be careful as these drawings will replace the HP EGS default arrowheads if they are stored back in their original volume or directory. If you want to save both your own and the default arrowheads, you can store the new arrowhead drawings in a separate volume or directory. Load the special arrowheads (using the <code>LOAD &lt;drawing&gt;</code> command of the Graphics Editor) before you edit the drawing.

Layers and levels      The levels in IGES correspond to the layers in HP EGS. If you choose to preserve layers, and if the level of a component in an IGES file is zero, unspecified, or greater than 255, the component is added to the default layer specified in the IGES-to-EGS default file.

## IGES-to-EGS Translator Process File

The IGES-to-EGS Translator extracts process file information such as units and layer definitions. If desired, this process file information can be extracted from the archive file, saved as an HP EGS process file (with the `_PR,TEXT` suffix), and loaded for subsequent editing of the drawing.

## Estimating Mass Storage Requirements and Run Time

The time needed to translate an IGES file is approximately 2 seconds per kilobyte of memory occupied by the IGES file and is directly related to the size of the file.

The HP EGS archive file produced by the translator is approximately one-fifth the size of the IGES file. If the size of the archive file exceeds the available space on your mass storage medium, the translator responds with the prompt:

```
Current output file full. Do you want to write the rest of
the output to another file? (Y/N)
```

If you type in Y, you will see:

```
Output destination:
```

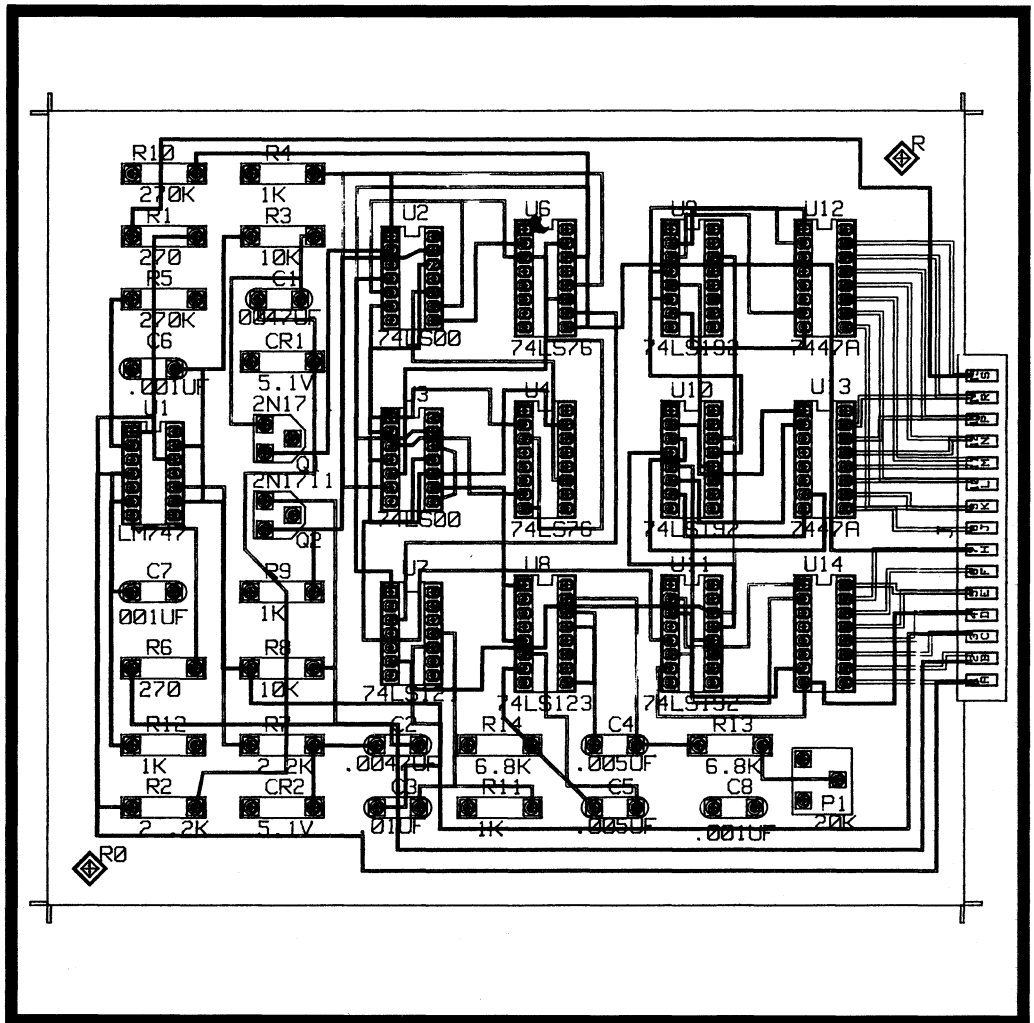
Enter a new file specification. If you type in N, the translation is aborted.



# HP EGS Application Note



## Using Archive Files to Transport Data



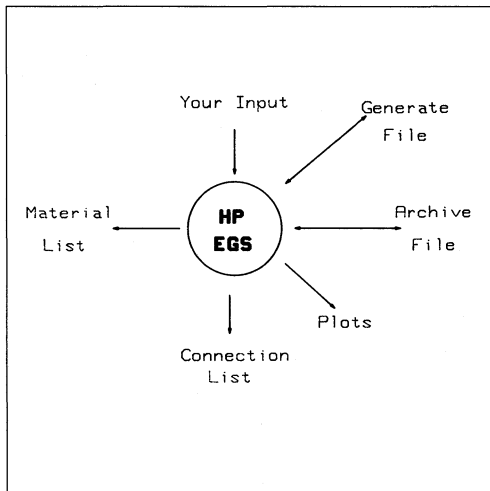
# HP EGS: The Open System

HP EGS is a computer-aided artwork tool that can help you prepare and use engineering drawings. But as you know, engineering drawings contain much more than geometrical design information; they also include identifying text, connectivity information, parts quantities, and manufacturing specifications. HP EGS provides several ways by which you can transfer this drawing information from the system to your particular design and manufacturing applications. In addition, HP EGS accepts information through several input formats.

This note has two purposes:

- To briefly explain the formats by which you can transfer information to and from HP EGS.
- To show you an example that uses a post-processing program to extract information from an HP EGS Archive file. As you follow the example, you'll see what an HP EGS Archive file looks like and how you can use it. Because this information focuses on a specific example, you will find that it contains some technical terms that are related to HP EGS and programming.

Here are some of the ways you can transfer data to and from HP EGS. The arrowheads indicate the possible directions of data flow.



HP EGS: The Open System

## Transferring Data to HP EGS

HP EGS accepts three types of data input. One type is your own input via the HP EGS command set. The other two types are Archive and Generate files.

## User Input

In addition to inputting information to HP EGS via its command set, you can use the INPUT command to stream files that contain HP EGS commands.

## Archive Files

HP EGS Archive files contain the commands necessary to reconstruct your drawing. Additionally, Archive files preserve the identity of drawings that are added to the main drawing. To illustrate this concept, consider a printed circuit (PC) board containing many 16-pin dual-inline packages (DIPs). You could add these DIPs to the drawing by constructing each one separately from basic components (such as rectangles, circles, and text). You would be more efficient, however, if you constructed one drawing of the DIP, saved it as a separate file, and then added the DIP drawing (with one command instead of many) to each location on the PC board.

You can create an HP EGS Archive file with the ARCHIVE command and bring it back into the system with the RETRIEVE command. RETRIEVE essentially uses the commands listed in the Archive file to reconstruct the drawing. Because the file is a list of commands, you can use the text editor of the system to modify the file and thus change the appearance of the drawing when it is retrieved.

## Generate Files

Like Archive files, Generate files contain the commands necessary to reconstruct your drawing. Generate files, however, do not preserve the identity of drawings added to the main drawing. Instead, the GENERATE command breaks all of the added drawings down to their basic components. Recall the previous 16-pin DIP example. For each DIP, the Archive file contains a reference to the drawing of the DIP. The Generate file, in contrast, builds the DIPs from basic components (such as circles, lines, and rectangles) by using HP EGS ADD commands.

You can modify or reformat Generate files in the same ways as Archive files.

## Transferring Data from HP EGS

HP EGS provides five types of data output. They are plot files, connection lists, material lists, Generate files, and Archive files.

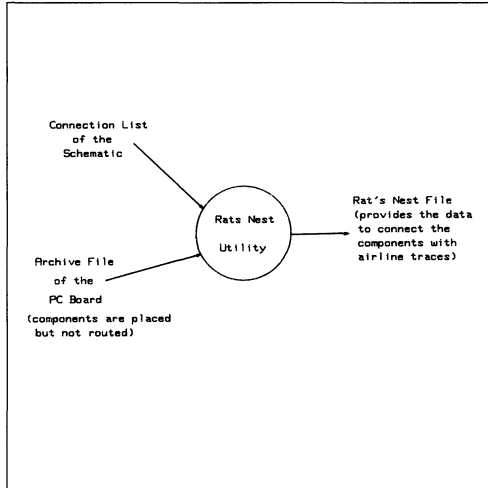
## Plot Files

The plotter is a relatively simple way to transfer information from HP EGS. With it you can obtain paper, mylar, or acetate copies of complete drawings or specify certain sections of a drawing for applications such as printed circuit (PC) board trace masking. In addition, HP TechWriter allows you to merge HP EGS drawings with text for office documentation. You can use dot-matrix printers for screen dumps of HP EGS drawings and both dot-matrix printers and the HP LaserJet for HP TechWriter text.



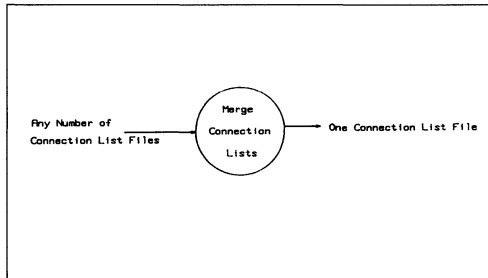
## Connection Lists

An HP EGS Connection List details logical or physical connectivity information. HP EGS offers three post-processors that take the information contained in a connection list and manipulate it for a particular application. One of these post-processors is the Rat's Nest Generator. Using connectivity information from a schematic drawing and the Archive file of a placed PC board, the generator draws airline (straight line as opposed to routed) traces between nets on a PC board. You can then stretch the traces or move them to other layers so they do not overlap.



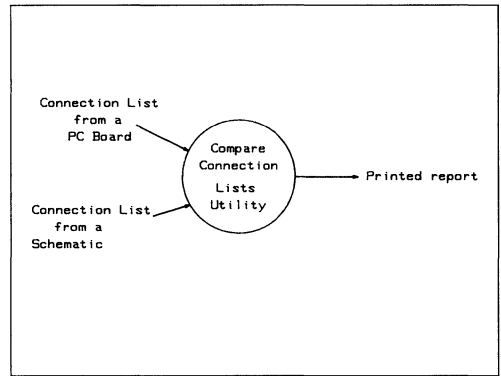
**HP EGS Rat's Nest Generator**

Another HP EGS post-processor that uses the connection list is the Merge Connection List utility. It merges several connection lists into one list.



**Merging Connection Lists**

The third HP EGS post-processor that uses connection lists is the Compare Connection List utility. It compares the connection lists from PC boards and schematics to ensure complete routing.

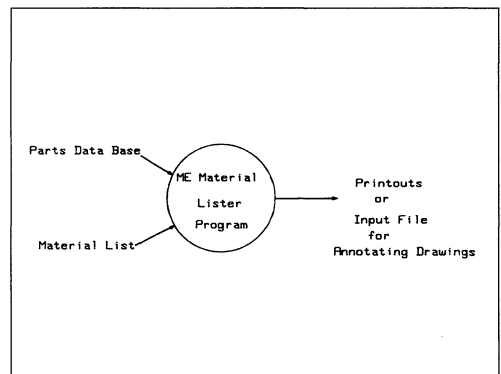


**Comparing Connection Lists**

Through programs you write, you can also reformat the information contained in the list to create net list outputs for logic simulation.

## Material Lists

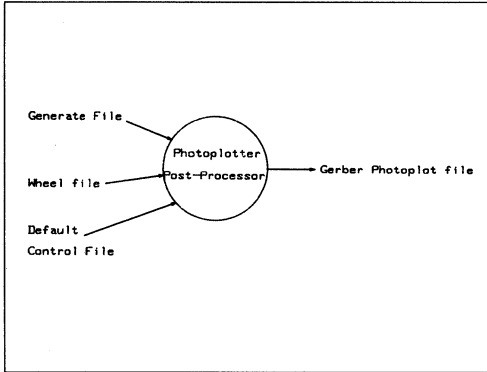
HP EGS offers a material listing program that can be used to detail the number and type of parts on either mechanical or electrical drawings. Drawing upon a data base specific to the application, the program produces both bill of material printouts and input files. You can use the input files to annotate your drawing with a bill of materials.



**The Material Lister**

## Generate Files

The HP EGS GENERATE command creates a Generate file. You can use a program to either extract information from the file or reformat it for a specific application. Two post-processors that reformat Generate file information are the HP 98310 HP EGS Photoplotter and N/C Drill Tape utilities. Although not illustrated, the N/C Drill Tape utility operates similarly to the Photoplotter shown below.

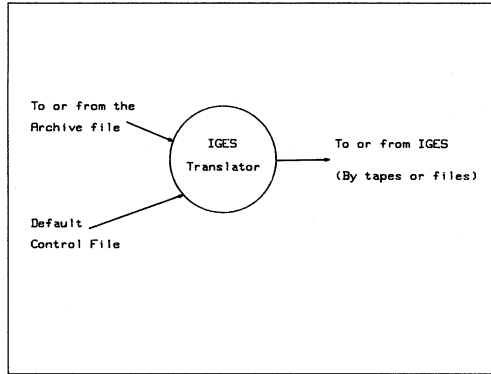


**The HP EGS Photoplot Utility**

The edge router is another computer-aided manufacturing application that combines Generate file information with command and control files to produce a drive output.

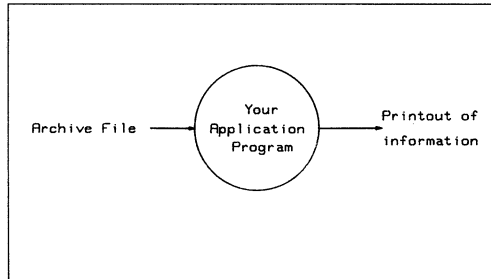
## Archive Files

To create an HP EGS Archive file, you use the ARCHIVE command. Like a Generate file, the information in the Archive file can be reformatted for use with a particular application. The HP 98311 HP EGS IGES Translator, for example, reads the information contained in the Archive file and reformats it to the Initial Graphics Exchange Specification (IGES) 1.0 standard.



**The IGES Translator**

You can also create and run your own application program to extract certain information from the Archive file.

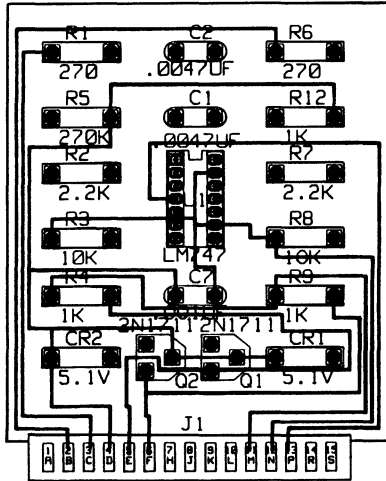


**Extracting Information With a Program**

The following sections explain this concept through an example. The example not only shows that a program can be run on an Archive file, but also that the Archive file can be used in different operating systems.

# Understanding the Archive File

We have seen that the HP EGS Archive file is both a file containing the commands necessary to reconstruct a drawing and a medium by which you can transfer data to and from HP EGS. Building from these ideas, the following sections develop an example that uses a post-processing program to find the total length of traces on the PC board shown below.



Example PC Board Drawing

Shorter traces would allow more traces per board side.  
The following sections develop the example and show you how to fully use the file by:

- Creating an Archive file.
- Looking at the file format.
- Archiving specific drawing information.
- Extracting file data with two programs – one written in Pascal and one in BASIC.

## Creating an Archive File

The HP EGS ARCHIVE command produces an Archive file\*. To Archive the example drawing which we will call board, you would enter:

```
ARC :P #3:board #4:;
```

(ARC is an abbreviation for ARCHIVE.) The file of the drawing named board is located in volume #3. The command specifies that HP EGS place the Archive file in volume #4. The system distinguishes the Archive file by calling it board.a. The ARCHIVE command has several options that allow you to specify or limit the drawing information that is placed in the Archive file. For example, notice the :P option included in the above command line. It specifies to begin the Archive file with the drawing process file. This file contains the drawing defaults such as units, the grid setting, and color and line type assignments.

## Looking at the File Format

The Archive file board.a appears on page 6. Remember that this is the file HP EGS reads when you invoke the RETRIEVE command. Because of the length of this example file, sections have been deleted where they do not illustrate any additional information. The deleted sections are indicated by a series of three dots.

\* You can find complete information about the ARCHIVE command in the HP EGS Syntax Reference (HP P/N 98305-90005).

```

## HP EGS Graphics Editor
## Revision      : 2.0
## Archival Date : 2/20/85
## Archival Time : 5:30 PM

```

```

UNITS MIL,2;
LOCK 90.0000;
GRID 25.0,4 0.0,0.0;

```

```

.
.
.
TNAME :L1 :F5.0 :T1004;
##      Layer Line
## Label   Type  Type    Color   Pen   Layer
EQU INSTBOUN :D    :L8    :C1    :P1    0;
EQU  PORILL  :D    :L1    :C6    :P1    1;
.
.
.
EQU DRILLTAG :D    :L1    :C4    :P0    56;
EQU  ERRORS  :D    :L1    :C1    :P1    255;
EXIT;

```

```

$FILES board, transpad, c300, dip14, lp40o, lp40r,
r450, pad46c, lp46c, edge156, ec156bb, ec156f $;

```

```

EDIT;
EDIT;

```

These lines identify the file as an HP EGS file and list the archival date and time. Note: Lines beginning with ## indicate comment lines and the system ignores them during retrieval.

Because :P was specified with the ARCHIVE command, the Process file is next. This file lists the drawing defaults at the time of archiving. You can modify these defaults or change layer designations with the Editor.

When HP EGS is retrieving a drawing, it uses the \$FILES command to remove any drawing files with the same name in the destination volume (the volume in which the Archive file is being placed). This prevents placement of two drawings with the same name in the same volume. Listed after \$FILES is the main drawing name and the name of any drawings contained in the main drawing. In this example board is the main drawing. The list then continues with the 11 drawings contained in board.

The two EDIT commands clear the CRT screen before the drawing is retrieved.

```

EDIT ec156f;
SHOW #E;
LOCK 0.0;
LEVEL 0;
GRID 0.5,1 0.0;
WINDOW -78.0,0.0 78.0,290.0;
ADD N53 :F25.0 :S0 :AD :T1001 "IGNORE" 50.0,180.0;
ADD L2 :W0.0 -78.0,0.0 78.0,0.0;
ADD R5 :W0.0 -40.0,70.0 40.0,290.0;
ADD R6 :W0.0 -40.0,70.0 40.0,290.0;
GRID 5.0,1 0.0,0.0;
LOCK 0.0000;
SAVE;
.
.
.

```

The other ten drawings contained in `board`, (`transpad`, `c300`, `dip14`, `lp40o`, `lp40r`, `r450`, `pad46c`, `lp46c`, `edge156`, and `ec156bb`), are each represented in the Archive file by similar series of EDIT commands. These series have been deleted from this example.

The final EDIT series lists the commands necessary to reconstruct the main drawing, place the 11 drawings within it, and add the trace lines.

```

.
.
.
EDIT board;
SHOW #E;
LOCK 0.0;
LEVEL 1;
GRID 0.5,1 0.0;
WINDOW -925.0,-1700.0 2025.0,1975.0;
ADD edge156 :ND -600.0,-1700.0;
    BEGIN_ASSOC
    ADD N51 :F50.0 :S0 :T1003 :D
        :L 1272.0,-1466.0 [5] "13" 1222.0,-1491.0;
    .
    .
    .
    END_ASSOC
.
.
.
ADD L5 :W15.0 :FILL 625.0,-850.0 250.0,-850.0
    250.0,-750.0 0.0,-750.0 0.0,-1125.0
    25.0,-1125.0 25.0,-1500.0;
GRID 25.0,4 0.0,0.0;
LOCK 90.0000;
SAVE;

```

For each drawing in the `$FILES` list, the Archive file contains a series of commands that begins with EDIT. In this case, the drawing `ec156f` is the first drawing retrieved. The ADD commands build the drawing by adding a note, a line, and two rectangles.

The BEGIN\_ASSOC and END\_ASSOC commands associate text to a component. In this case, the number 13 was associated to the `edge156` component.

The GRID and LOCK commands reflect the system settings when the drawing was archived.

## Archiving Specific Information

While skimming over the previous Archive file you might have noticed the ADD L5 command near the end of the file. This command, and the other ADD L5 commands that have been deleted, add the traces to board. To count the total length of traces on this PC board, a program needs to calculate the distances specified by the coordinates of each line. The criteria used by the program to find and interpret the ADD L5 commands can be simplified by using ARCHIVE command options to eliminate everything else from the file. The following command does this.

```
ARC -E +L5 #3:board #4: ;
```

This command uses two options. The first, -E, specifies that the Archive file contains no components. If this was the only option in the command line, you would get a blank drawing after retrieving the Archive file. The other option, +L5, specifies to include all lines on layer 5. As before, the drawing board resides in volume #3 and the Archive file, board\_a, is placed in volume #4.

The archive file appears as:

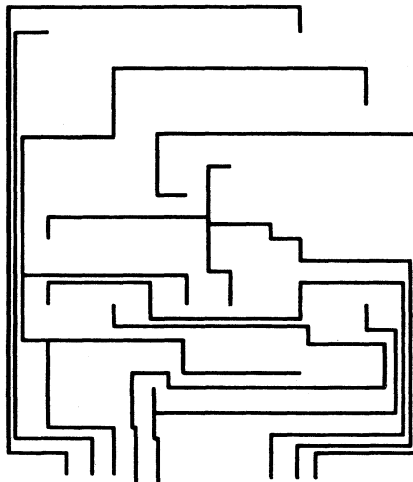
```
## HP EGS Graphics Editor
## Revision      : 2.0
## Archival Date : 2/20/85
## Archival Time : 9:05 PM

$FILES board $;

EDIT;
EDIT;

EDIT board;
SHOW #E;
LOCK 0,0;
LEVEL 1;
GRID 0,5,1 0,0;
WINDOW -925,0,-1700,0 2025,0,1975,0;
ADD L5 :W15,0 :FILL 850,0,-750,0
      1150,0,-750,0;
.
.
ADD L5 :W15,0 :FILL 350,0,-750,0
      350,0,-525,0 -575,0,-525,0
      -575,0,-750,0;
ADD L5 :W15,0 :FILL 625,0,-850,0
      250,0,-850,0 250,0,-750,0
      0,0,-750,0 0,0,-1125,0
      25,0,-1125,0 25,0,-1500,0;
GRID 25,0,4 0,0,0,0;
LOCK 90,0000;
SAVE;
```

When this Archive file is retrieved, the drawing appears as:



Traces of board

## Extracting Data With Two Programs

Two programs – one in Pascal and one in BASIC – appear in the appendices of this note. Each reads the Archive file and counts the total length of traces on the drawing board.

Because HP EGS operates on the Pascal operating system, you can write a Pascal program, compile it\*, and execute it without leaving the operating system. To use the Archive file while in the BASIC operating system, however, you must first enter the Pascal Filer and translate the Archive file from Pascal Data format to ASCII. After translating the file, remove the .ASC from the file name. You can then boot up in BASIC and run your BASIC program on it to calculate the total length of traces.

\* Not all versions of HP EGS include a Pascal compiler. Check with an HP sales representative.

# In Summary. . .

---

---

Throughout this note, you have learned:

- The formats by which you can transfer information to and from HP EGS. You have seen several applications that use these formats.
- How to create an Archive file.
- The Archive file format.
- How to archive specific drawing information.
- How to extract data from the Archive file using programs from different operating systems.

No doubt you have already thought of some ways you can pull data from HP EGS drawings to use with your own applications. Take a few minutes to discuss these ideas with your HP EGS Product Specialist. With his or her knowledge and the support and documentation that can be obtained with HP EGS, you can extend the system to meet your engineering and production demands.

## Other Sources of Information

The following documents contain more information that can help you use HP EGS.

<b>Subject</b>	<b>Information Source</b>	<b>Part Number</b>
HP EGS Overview	HP EGS Sales Brochure	5953-9522
HP EGS Tutorial	<i>Learning HP EGS</i>	98305-90003
HP EGS Theory and Operation	<i>Understanding HP EGS</i>	98305-90004
HP EGS Commands and Archive Reference	<i>HP EGS Syntax Reference</i>	98305-90005
Upgrading from EGS/200 to HP EGS 2.0	<i>HP EGS Upgrade Note</i>	98305-90007
HP TechWriter	Data Sheet	5953-9521
HP TechWriter Tutorial and Operation	<i>HP TechWriter User's Guide</i>	98819-90000
HP TechWriter Commands and Reference	<i>HP TechWriter Reference</i>	98819-90001

# Appendix A: The Pascal Program

---

---

The following Pascal program counts the total length of traces on the drawing board. After you have compiled the program\*, execute it, and a prompt will ask for the name of the Archive file. After you type in the file name,

the program looks at the coordinates of all of the lines and computes the distances between these coordinates. For the example PC drawing, `board`, the total length of traces is 47,350 mils (47.35 inches).

```
#DEBUG ON#
#SWITCH_STRPOS ON#
#SYSPROG ON#
PROGRAM llength(INPUT, OUTPUT);

TYPE
  string255          =STRING[255];

VAR
  total_length      :REAL;
  inline            :string255;
  archive_file      :TEXT;
  archive_file_name :string255;
  found-it          :BOOLEAN;
  x1                 :REAL;
  y1                 :REAL;
  x2                 :REAL;
  y2                 :REAL;

PROCEDURE calculate_length;

VAR
  i : INTEGER;

BEGIN
  STRODELETE ( inline, 1, 5 );

  found-it := FALSE;
  WHILE ( NOT found-it ) DO
    BEGIN
      TRY
        STRREAD ( inline, 1, i, x1 );
        STRREAD ( inline, i+1, i, y1 );
        found-it := TRUE;
      RECOVER
        BEGIN
          i := STRPOS ( inline, ' ' );
          STRODELETE ( inline, 1, i );
        END;
    END;

    WRITELN ( x1, ',', y1 );
    STRODELETE ( inline, 1, i );

    found-it := FALSE;
    WHILE ( NOT found-it ) DO
```

The program finds all ADD L5 commands and then deletes the ADD L5.

Finds first (x,y) pair.

Deletes first (x,y) pair.

\* Not all versions of HP EGS include a Pascal compiler. Check with an HP sales representative.



```

BEGIN
    STRREAD(inline, 1, i, x2);
    STRDELETE(inline, 1, i);
    STRREAD(inline, 1, i, y2);
    WRITELN(x2, ', ', y2);
    total-length := total-length + SQRT((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
    x1 := x2;
    y1 := y2;
    IF i > STRLEN(inline) THEN
        READLN(archive-file, inline)
    ELSE
        IF STR(inline, i, 1) = ';' THEN
            found-it := TRUE
        ELSE
            STRDELETE(inline, 1, i);
    END;
    WRITELN('End-Of-Line');
END ;

```

Gets all x,y for this line.

ADD command spans multiple file lines.

Removes this x,y.  
Gets all x,y for this line.

Ends the PROCEDURE calculate-length.

```

BEGIN
WRITE('Enter the name of the archive file: ');
READLN(archive-file-name);
RESET(archive-file, archive-file-name, 'SHARED');

total-length := 0.0;
WHILE (NOT EOF(archive-file)) DO
    BEGIN
        READLN ( archive-file, inline );
        IF STRLEN ( inline ) >= 6 THEN
            IF STR ( inline, 1, 6 ) = 'ADD L5' THEN calculate-length;
    END ;
    WRITELN('Total length of traces is: ', total-length);
CLOSE(archive-file);
END.

```

Begins the program length.  
Asks for the file name.

Adds up the lengths.

# Appendix B: The BASIC Program

---

---

This BASIC program calculates the total length of traces on the drawing board. It was written while in the BASIC 3.0 operating system. Unlike the previous Pascal program that looked only for ADD L5 commands, this

BASIC program can look through a complete Archive file (such as the first one shown in this note) and pick out the ADD L5 commands.

```
10 !
20 ! PROGRAM LLENGTH
30 !
40 !
50 DIM Archive#[255],A_data#[255],D_name#[255]
60 !
70 Error_place=0
80 O_device=701
90 !
100 !
110 ! Get the archive file name and open the archive file
120 !
130 INPUT "Enter the name of the archive file",Archive#
140 !
150 ON ERROR GOTO Read_error
160 ASSIGN @Archive TO Archive#
170 OFF ERROR
180 !
190 !
200 ! Find the string '#FILES' in the archive file
210 !
220 Error_place=1
230 Found_it=0
240 WHILE (NOT Found_it)
250 ON ERROR GOTO Read_error
260 ENTER @Archive;A_data#
270 OFF ERROR
280 IF LEN(A_data#)>>8 THEN
290 IF A_data#[1,7]="#FILES " THEN
300 Found_it=1
310 D_name#=A_data#[8,6+POS(A_data#[8,LEN(A_data#)])," "]
320 OUTPUT O_device;"Drawing name is: ";D_name#;"."
330 END IF
340 END IF
350 END WHILE ! Find '#FILES'
360 !
370 !
380 ! Find the string 'EDIT <drawing name>' in the archive file
390 !
400 Error_place=2
410 Found_it=0
420 WHILE (NOT Found_it)
430 ON ERROR GOTO Read_error
440 ENTER @Archive;A_data#
450 OFF ERROR
460 IF A_data#="EDIT "&D_name#&";" THEN
470 Found_it=1
480 END IF
490 END WHILE ! Find 'EDIT <drawing name>'
500 !
510 !
```

```

520 ! Find the total length of all lines in the drawing O_name#
530 !
540 Error_place=3
550 False=0
560 WHILE (NOT False)
570   ON ERROR GOTO Read_error
580   ENTER @Archive;A_data#
590   OFF ERROR
600   IF LEN(A_data#)<5 THEN 1140
610   IF A_data#[1,5]<>"ADD L" THEN 1140
620   A_data#*=A_data#[6,LEN(A_data#)]
630   GOSUB Next_word
640   !
650   !
660   ! Find the first X,Y pair
670   !
680   Found_it=0
690   WHILE (NOT Found_it)
700     ON ERROR GOTO Value_error
710     X1=VAL(A_data#)
720     OFF ERROR
730     GOSUB Next_word
740     Y1=VAL(A_data#)
750     GOSUB Next_word
760     Found_it=1
770   END WHILE ! Find first X,Y pair
780   !
790   OUTPUT O_device;X1;" , ";Y1
800   !
810   !
820   ! Get all X,Y pairs for this line
830   !
840   Found_it=0
850   WHILE (NOT Found_it)
860     X2=VAL(A_data#)
870     GOSUB Next_word
880     Y2=VAL(A_data#)
890     GOSUB Next_word
900     OUTPUT O_device;X2;" , ";Y2
910     Total_length=Total_length+SQR((X1-X2)^2+(Y1-Y2)^2)
920     X1=X2
930     Y1=Y2
940     IF A_data#="" THEN
950       ON ERROR GOTO Read_error
960       ENTER @Archive;A_data#
970       OFF ERROR
980       !
990       !
1000      ! Delete all leading blanks and commas from new line
1010      !
1020      Done=0
1030      WHILE A_data#<>" " AND (NOT Done)
1040        IF A_data#[1,1]=" " OR A_data#[1,1]="," THEN GOSUB Next_char
1050        IF A_data#="" THEN 1070
1060        IF A_data#[1,1]<>" " AND A_data#[1,1]<>"," THEN Done=1
1070      END WHILE ! Delete leading blanks and commas
1080      ELSE
1090        IF A_data#[1,1]="," THEN Found_it=1
1100      END IF

```

```

1110     END WHILE      ! Get all X,Y pairs for this line
1120     !
1130     OUTPUT D_device;"End-Of-Line"
1140 END WHILE ! Total length of all lines
1150 !
1160 !
1170 ! Delete the first character of A_data$
1180 !
1190 Next_char: !
1200     IF LEN(A_data$)=1 THEN
1210         A_data$=""
1220     ELSE
1230         A_data$=A_data$[2,LEN(A_data$)]
1240     END IF
1250     RETURN
1260 !
1270 !
1280 ! Delete the first word, and the following separator(s), in A_data$
1290 ! Words consist of any number of all characters, except for
1300 ! the characters that define the separators.
1310 ! Separators consist of the following characters:
1320 !     A space      - " "
1330 !     A comma      - ","
1340 !     A semi-colon - ";"
1350 !
1360 Next_word: !
1370     Done=0
1380     WHILE A_data$[<>]" AND (NOT Done)
1390         IF A_data$[1,1]<>" " AND A_data$[1,1]<>"," AND A_data$[1,1]<>";"
THEN GOSUB Next_char
1400         IF A_data$="" THEN 1420
1410         IF A_data$[1,1]=" " OR A_data$[1,1]="," OR A_data$[1,1]=";" THEN
Done=1
1420         END WHILE
1430         Done=0
1440         WHILE A_data$[<>" AND (NOT Done)
1450             IF A_data$[1,1]=" " OR A_data$[1,1]="," THEN GOSUB Next_char
1460             IF A_data$="" THEN 1480
1470             IF A_data$[1,1]<>" " AND A_data$[1,1]<>"," THEN Done=1
1480         END WHILE
1490         RETURN
1500 !
1510 !
1520 ! Handle 'VAL' errors here
1530 !
1540 Value_error: !
1550     OFF ERROR
1560     !
1570     IF ERRN=32 THEN
1580         GOSUB Next_word
1590         GOTO 700
1600     ELSE
1610         PRINT "Error Number ";ERRN;" occurred trying to find the first
X,Y pair."
1620         BEEP
1630     END IF
1640     !
1650     GOTO End_of_program

```

```

1660 !
1670 !
1680 ! Handle 'ENTER' errors here
1690 !
1700 Read_error: !
1710             OFF ERROR
1720             !
1730             IF ERRN=59 THEN
1740                 SELECT Error_place
1750                     CASE 1
1760                         OUTPUT 0_device;"#FILES ' statement not found in archive
file ";Archive$;"',"
1770                         DISP "#FILES ' statement not found in archive file
";Archive$;"',"
1780                         BEEP
1790                     CASE 2
1800                         OUTPUT 0_device;"'EDIT <drawing name>' statement not found
in archive file ";Archive$;"',"
1810                         DISP "'EDIT <drawing name>' statement not found in archive
file ";Archive$;"',"
1820                         BEEP
1830                     CASE ELSE
1840                         OUTPUT 0_device;"Total length of lines is: ";Total_length
1850                         DISP "Total length of lines is: ";Total_length
1860                     END SELECT
1870             ELSE
1880                 OUTPUT 0_device;"Error Number ";ERRN;" occurred trying to read
archive file ";Archive$;"',"
1890                 DISP "Error Number ";ERRN;" occurred trying to read archive file
";Archive$;"',"
1900                 BEEP
1910             END IF
1920             !
1930             ASSIGN @Archive
1940             !
1950 End_of_program: END ! LLENGTH

```



# Subject Index

## a

Aborting commands while in the text editor .....	<b>M 31</b>
Access .....	<b>M 128</b>
Activating tablet menus .....	<b>L 285, S 197, U 90</b>
Active:	
Commands .....	<b>S 5</b>
Drawing area .....	<b>U 86</b>
ACTUAL:	
DRILL (Photoplot/Drill) .....	<b>U 207</b>
PHOTOPLOT (Photoplot/Drill) .....	<b>U 207</b>
AdBox .....	<b>L 55</b>
AdCom .....	<b>L 55</b>
ADD .....	<b>S 7</b>
ADD (archive) .....	<b>S 244-245</b>
ADD ARC .....	<b>S 10</b>
ADD ARC (archive) .....	<b>S 246</b>
ADD CIRCLE .....	<b>S 14</b>
ADD CIRCLE (archive) .....	<b>S 247</b>
ADD DIMENSION .....	<b>S 18</b>
ADD DIMENSION (archive) .....	<b>S 248-249</b>
ADD HATCH .....	<b>S 24-28</b>
ADD HATCH (archive) .....	<b>S 250-251</b>
ADD instance .....	<b>S 30-31</b>
ADD INSTANCE (archive) .....	<b>S 252-253</b>
ADD LINE .....	<b>S 32-37</b>
ADD LINE (archive) .....	<b>S 254</b>
ADD MARKER .....	<b>S 39</b>
ADD MARKER (archive) .....	<b>S 255</b>
ADD NOTE .....	<b>S 48-52</b>
ADD NOTE (archive) .....	<b>S 260-261</b>

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 . . . . .	<b>S 244-296</b>
Description . . . . .	<b>S 236-242, U 24-27</b>
\$FILES . . . . .	<b>S 239</b>
Process file in . . . . .	<b>S 236</b>
Retrieving drawings and instances with EDIT . . . . .	<b>S 239</b>
Retrieving macro instances with DEF :I . . . . .	<b>S 240</b>
Using to transport data . . . . .	<b>U 259</b>
Archive Reference . . . . .	<b>S 235-296</b>
Archived drawings, retrieving with RETRIEVE . . . . .	<b>S 192</b>
Archiving drawings . . . . .	<b>U 137</b>
AREA . . . . .	<b>S 56-57</b>
AREA, isometric . . . . .	<b>S 57</b>
AREA? . . . . .	<b>L 53</b>
Array:	
Creating with STEP . . . . .	<b>S 204-207</b>
Step specification . . . . .	<b>S 8</b>
ASCII ordinals . . . . .	<b>U 98</b>
Assembler . . . . .	<b>M 73</b>
Assembling parts . . . . .	<b>L 132</b>
Assembly drawing:	
Annotating . . . . .	<b>L 147</b>
Creating . . . . .	<b>L 268</b>
Material listing . . . . .	<b>L 156</b>
Associated text:	
:AC text . . . . .	<b>S 50, U 20-21, U 136</b>
:AD text . . . . .	<b>S 50, U 20-21, U 136</b>
Description . . . . .	<b>U 20-21</b>
Drawing . . . . .	<b>U 78, U 87</b>
Tag values . . . . .	<b>U 136, U 197</b>
Attribute data . . . . .	<b>U 171, U 203-206</b>
ATTRIBUTES (Photoplot/Drill) . . . . .	<b>U 203</b>
AUTOFLASH (Photoplot/Drill) . . . . .	<b>U 187, U 206</b>

# 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 . . . . .	<b>L 116</b>
Library parts . . . . .	<b>U 77-84</b>
Macros with DEFINE . . . . .	<b>S 80-85</b>
Material list . . . . .	<b>L 209</b>
Material list data . . . . .	<b>L 156</b>
Parts list . . . . .	<b>L 159</b>
Schematic library part . . . . .	<b>L 183</b>
CRT modules for the INITLIB file. . . . .	<b>I 68</b>
CTRL BACK SPACE . . . . .	<b>L 6</b>
CTRL CLR LN. . . . .	<b>L 6</b>
Current:	
Command . . . . .	<b>U 59</b>
Working directory . . . . .	<b>M 11</b>
CURSOR . . . . .	<b>S 72-75, U 16, U 22</b>
Cursor:	
In text editor . . . . .	<b>M 27, M 32</b>
Moving . . . . .	<b>L 13</b>
Size . . . . .	<b>S 5, S 72-74</b>
Snapping modes. . . . .	<b>L 17</b>
User-defined . . . . .	<b>S 72-75</b>
Customizing:	
Steps . . . . .	<b>U 35</b>

# 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 .....	<b>L 124, S 18-22</b>
Dimension values:	
Font size .....	<b>S 92</b>
Number of decimal places .....	<b>S 88</b>
Orientation .....	<b>S 89</b>
Radix .....	<b>S 93</b>
Scale .....	<b>S 95</b>
Slant .....	<b>S 97</b>
Units .....	<b>S 98-99</b>
DIMENSION, ADD .....	<b>S 18</b>
*DIMN .....	<b>L 103</b>
DIM_DEC .....	<b>S 88</b>
DIM_DEC (archive) .....	<b>S 266</b>
DIM_DIR .....	<b>S 89</b>
DIM_DIR (archive) .....	<b>S 267</b>
DIM_EXTEND .....	<b>S 90-91</b>
DIM_EXTEND (archive) .....	<b>S 268</b>
DIM_FSIZE .....	<b>S 92</b>
DIM_FSIZE (archive) .....	<b>S 269</b>
DIM_RADIX .....	<b>S 93</b>
DIM_RADIX (archive) .....	<b>S 270</b>
DIM_SCALE .....	<b>S 95</b>
DIM_SCALE (archive) .....	<b>S 271</b>
DIM_TFSIZE .....	<b>S 96</b>
DIM_TFSIZE (archive) .....	<b>S 272</b>
DIM_TSLANT .....	<b>S 97</b>
DIM_TSLANT (archive) .....	<b>S 273</b>
DIM_UNITS .....	<b>S 98-99</b>
DIM_UNITS (archive) .....	<b>S 274-275</b>
Directories, listing .....	<b>M 55, M 144</b>
Disc drive, finding the drive number .....	<b>I 15</b>
DISPDISP .....	<b>L 233</b>
DISPLAY .....	<b>S 100-101</b>
Display:	
Grid, setting with GRID .....	<b>S 135-137, U 14-15</b>
Optimization layers .....	<b>L 231</b>
Option switches, setting on a video interface card .....	<b>I 19</b>
Setting the default with the Codeword utility .....	<b>I 46</b>
DISPPEN .....	<b>L 233</b>
DISTANCE .....	<b>S 102</b>
DIBox .....	<b>L 55</b>

DICom .....	L 55
Delete .....	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 . . . . .	<b>U</b> 131-135
EWEE on a hard disc . . . . .	<b>I</b> 30-31
EWEE on an SRM. . . . .	<b>I</b> 41-42
EWME on a hard disc . . . . .	<b>I</b> 30-31
EWME on an SRM . . . . .	<b>I</b> 41-42
EWSYS on a hard disc . . . . .	<b>I</b> 29
EWSYS on an SRM . . . . .	<b>I</b> 40
Hardware needed for . . . . .	<b>I</b> 7-9
HP EGS codeword . . . . .	<b>I</b> 42-49
HP EGS on a new hard disc. . . . .	<b>I</b> 26-31
HP EGS on an SRM. . . . .	<b>I</b> 37-42
HP EGS on top of Pascal 3.1. . . . .	<b>I</b> 67-69
HP EGS Option 085. . . . .	<b>I</b> 23
HP EGS Options 042 and 044 . . . . .	<b>I</b> 23-42
HP EGS Software . . . . .	<b>I</b> 7
IGES Translator. . . . .	<b>U</b> 231-236
Pascal 3.1 alongside HP EGS 2.1 . . . . .	<b>I</b> 70
Photoplot/Drill . . . . .	<b>U</b> 172-178
Stream files . . . . .	<b>I</b> 27
Terms used . . . . .	<b>I</b> 5
Verifying . . . . .	<b>I</b> 50-52
Instance-in-Context EDIT. . . . .	<b>S</b> 108-111
Instance-oriented connection list. . . . .	<b>U</b> 148-156
Instance:	
ADD . . . . .	<b>S</b> 30-31
Adding. . . . .	<b>L</b> 63
Copying. . . . .	<b>L</b> 68
Creating with WRAP . . . . .	<b>S</b> 230-233
Description of . . . . .	<b>U</b> 28-31, <b>U</b> 126
Editing with EDIT. . . . .	<b>S</b> 107-108
Saving with SAVE . . . . .	<b>S</b> 196-198
Smashing. . . . .	<b>L</b> 69
Used to create library parts. . . . .	<b>U</b> 78
Viewing . . . . .	<b>L</b> 182
Interconnect:	
Layer, defining. . . . .	<b>S</b> 113
Layer type . . . . .	<b>U</b> 19
Mode, setting display. . . . .	<b>S</b> 227
Interface card, video. . . . .	<b>I</b> 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 .....	<b>M 15</b>
Largest space on a disc .....	<b>S 63</b>
Layer:	
Definitions .....	<b>L 52, L 99, L 224, U 52</b>
Description .....	<b>U 18-19</b>
Detail .....	<b>U 19</b>
Interconnect .....	<b>U 19</b>
Label .....	<b>U 52</b>
Number .....	<b>U 52</b>
Symbolic .....	<b>U 19</b>
Using while creating library parts .....	<b>U 77</b>
Layers, defining with EQUATE .....	<b>S 113</b>
*LEAD .....	<b>L 26, L 94</b>
Leader line option of ADD LINE .....	<b>S 33</b>
Leader lines, adding .....	<b>L 151</b>
LEVEL .....	<b>S 151, U 52</b>
LEVEL (archive) .....	<b>S 286</b>
LEVEL* .....	<b>L 177, L 232</b>
Levels, nesting .....	<b>L 66</b>
Librarian .....	<b>M 73</b>
Library parts:	
Creating a schematic .....	<b>L 183, U 77-84</b>
Description .....	<b>U 17</b>
Drill holes (Photoplot/Drill) .....	<b>U 223</b>
Flash apertures (Photoplot/Drill) .....	<b>U 219-220</b>
Ignore and no fill tags (Photoplot/Drill) .....	<b>U 222</b>
Macro instances .....	<b>U 78-82</b>
PC board .....	<b>L 238, L 278</b>
Polygon, hatch, rectangle fill aperture (Photoplot/Drill) .....	<b>U 220-222</b>
Schematic .....	<b>L 218</b>
Special considerations .....	<b>U 77-78</b>
Text/note draw aperture (Photoplot/Drill) .....	<b>U 222</b>
LINE .....	<b>L 28, L 167</b>

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 .....	<b>M 16</b>
*MAKE .....	<b>L 55</b>
Make .....	<b>M 146</b>
Make_Prt .....	<b>L 176, L 232</b>
Making connection lists .....	<b>L 198, L 264</b>
Making material lists .....	<b>L 198</b>
Manager menu, explanation of .....	<b>I 50-51</b>
Manufacture and assembly layers .....	<b>L 228</b>
Margin .....	<b>M 46, M 106</b>
MARKER, ADD .....	<b>S 39</b>
Mass storage, IGES Translator requirements .....	<b>U 249, U 257</b>
Master keepout layer .....	<b>L 230</b>
Material list data, creating .....	<b>L 156</b>
Material list:	
Adding .....	<b>L 162</b>
Creating .....	<b>L 198</b>
Description .....	<b>U 103</b>
Error checking .....	<b>U 114, U 123</b>
File, preparing with LIST .....	<b>S 153</b>
Of an assembly drawing .....	<b>L 156</b>
Output format .....	<b>U 117</b>
Parts file editor .....	<b>U 105</b>
Mathematical:	
Calculations with EVALUATE .....	<b>S 116</b>
Expressions .....	<b>U 16</b>
Expressions, creating with DEFINE .....	<b>S 80-85</b>
Matrix, creating radial matrix with STEP .....	<b>S 204-207</b>
*MDFY .....	<b>L 55</b>
Mechanical Engineering:	
Drawings .....	<b>L 1</b>
Personality .....	<b>L 97</b>
Personality file .....	<b>M 23</b>
Personality, installation on a hard disc .....	<b>I 30-31</b>
Personality, installation on an SRM .....	<b>I 41-42</b>
Personality loading .....	<b>L 104</b>
Post-processor .....	<b>U 103, U 115</b>
Tablet menu .....	<b>L 172</b>
Mechanical library parts .....	<b>L 168</b>
MEMORY .....	<b>S 163</b>
Memory:	
Requirements for HP EGS .....	<b>I 7-9</b>
Used by loading commands .....	<b>S 157</b>

Memvol.....	<b>M</b> 76
MENU.....	<b>S</b> 164-166, <b>U</b> 57
Menu file, loading with LOAD.....	<b>S</b> 156
Menu:	
Height.....	<b>U</b> 59
Horizontal or vertical.....	<b>U</b> 59
Items.....	<b>L</b> 16
Loading with LOAD.....	<b>S</b> 155-157
Manager.....	<b>I</b> 50-51
Modifying and moving screen menus with MENU.....	<b>S</b> 164-166
Row.....	<b>U</b> 58
Saving tablet menus with SAVE TMENU.....	<b>S</b> 197
Toggling on tablet menus with TMENU.....	<b>S</b> 217
Width.....	<b>U</b> 59
Merging connection lists.....	<b>U</b> 161-164
Message file:	
Customizing (Connection Lister).....	<b>U</b> 159-160
Customizing (Photoplot/Drill).....	<b>U</b> 216-218
Editing and indexing.....	<b>U</b> 94
Loading with LOAD HELP.....	<b>S</b> 156
Modifying.....	<b>U</b> 93-96
Message tags.....	<b>U</b> 93
*MIR.....	<b>L</b> 56
Mirroring of notes and text.....	<b>S</b> 49
mm>in.....	<b>L</b> 102
Model number, finding at boot-up.....	<b>I</b> 22
Modes.....	<b>L</b> 15
MODIFY.....	<b>S</b> 168-171
Modifying:	
Assembly.....	<b>L</b> 144
Drawing or instance.....	<b>S</b> 106-111
Macro file.....	<b>U</b> 62
Message file.....	<b>U</b> 93
Multiple components.....	<b>L</b> 54
Notes and text.....	<b>L</b> 196, <b>L</b> 251
Parts file.....	<b>U</b> 108
Parts screen menus.....	<b>L</b> 287
Process file.....	<b>U</b> 52
Tablet menu.....	<b>L</b> 287, <b>U</b> 85
Window.....	<b>L</b> 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
----------------------------------------------------	------

PHOTO PLOT FORMAT (Photoplot/Drill). . . . .	U 208
----------------------------------------------	-------

PHOTO PLOT ORIGIN (Photoplot/Drill) . . . . .	U 210
-----------------------------------------------	-------

PHOTO PLOT 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 .....	<b>M</b> 155
SAVE .....	<b>S</b> 196-198, <b>U</b> 52, <b>U</b> 62
SAVE (archive) .....	<b>S</b> 290
Saving a drawing .....	<b>L</b> 36, <b>L</b> 45, <b>U</b> 52, <b>U</b> 62
*SCAL .....	<b>L</b> 56
SCALE .....	<b>L</b> 38
Scale factor, changing with MODIFY .....	<b>S</b> 169
Schematic:	
Library parts .....	<b>L</b> 218
Tablet menu .....	<b>L</b> 222
Schematics .....	<b>U</b> 161
Scientific functions, calculating with EVALUATE .....	<b>S</b> 117
Screen dump:	
Initiating with PAGE .....	<b>S</b> 176
With PLOT .....	<b>S</b> 181
Screen menu:	
Building .....	<b>U</b> 57
Description .....	<b>L</b> 15, <b>U</b> 7
File structure .....	<b>U</b> 58
Modifying with MENU .....	<b>S</b> 164-166
Saving with SAVE .....	<b>S</b> 196-198
SEARCH .....	<b>S</b> 200-201, <b>U</b> 50
Search file:	
Building .....	<b>U</b> 50-51
Listing contents with LIST .....	<b>S</b> 153
Loading with LOAD .....	<b>S</b> 156
Saving with SAVE .....	<b>S</b> 196-198
Searching for patterns in text editor .....	<b>M</b> 42, <b>M</b> 99
Secondary commands .....	<b>U</b> 10-12
Select .....	<b>L</b> 23
Select codes, setting on a peripheral .....	<b>I</b> 16-17
Selecting apertures with PHOTOPLOT .....	<b>U</b> 188-193
Selecting menu items .....	<b>L</b> 16
*SEMI .....	<b>L</b> 40
Semicircle and tangent lines, adding .....	<b>L</b> 106
Serial number, finding with the Codeword utility .....	<b>I</b> 45
Set .....	<b>M</b> 43, <b>M</b> 115
Shape of notes and text .....	<b>S</b> 51

Shape, stretching . . . . .	L 32
Shared Resource Manager, <i>see</i> 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 .....	<b>S 18-22</b>
Tooling marks, adding .....	<b>L 262</b>
*Top30 .....	<b>L 82</b>
Traces:	
Adding .....	<b>L 261</b>
Connecting .....	<b>L 257, L 271</b>
Tracing or flashing .....	<b>U 181</b>
Transferring files:	
From magnetic/paper tape .....	<b>M 16</b>
To magnetic/paper tape .....	<b>M 16</b>
Transformations with PHOTOPLOT .....	<b>U 186</b>
Translate .....	<b>M 61, M 157</b>
TSLANT .....	<b>S 220</b>
TSLANT (archive) .....	<b>S 293</b>
Type-ahead buffer .....	<b>U 46</b>
Typing over characters in text editor .....	<b>M 36, M 121</b>

## u

UNDO* .....	<b>L 23</b>
Unfilling an item. ....	<b>L 50</b>
Ungathering a collection. ....	<b>S 123-126</b>
Unit directory .....	<b>M 159</b>
Unit number (disc drive number) .....	<b>I 5</b>
Units .....	<b>M 5</b>
UNITS .....	<b>S 222, U 12-13, U 53</b>
UNITS (archive) .....	<b>S 294-295</b>
Unused space on a disc .....	<b>S 63</b>
Unwrapping instances with SMASH .....	<b>S 203</b>
Updating a title block .....	<b>L 129, L 153</b>
User restart .....	<b>M 84</b>
User:	
Grid .....	<b>L 18, U 14-15</b>
Units .....	<b>U 12-15, U 53</b>
Written applications .....	<b>U 46</b>

# 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
*WDTH .....	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
Xchng. ....	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