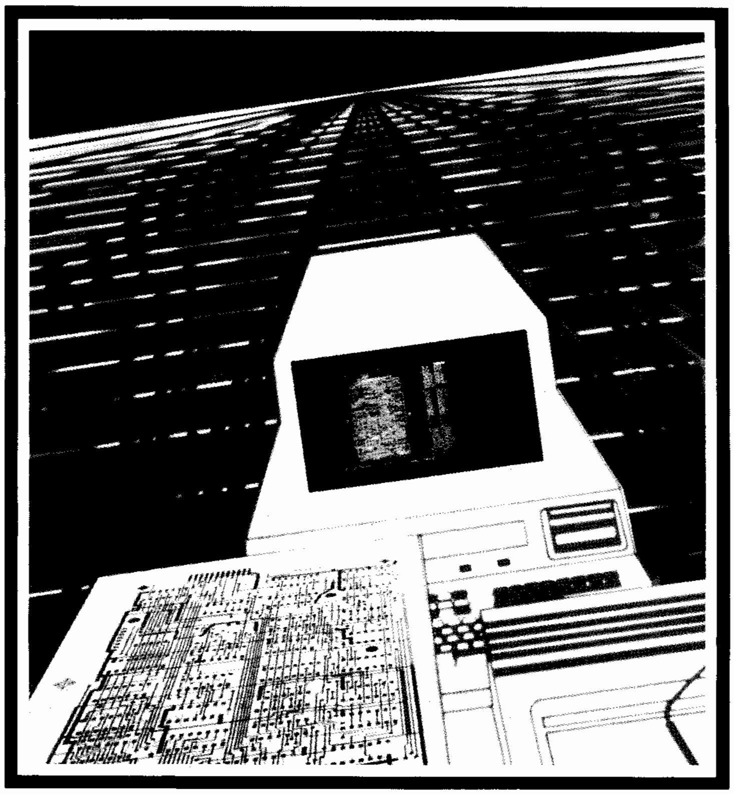
Engineering Graphics System

For the HP 9845



HEWLETT PACKARD

Engineering Graphics System

for the HP 9845

Manual Part No. 98301-12111
Cartridge Tape Part No. 98301-13754
Cartridge Tape Part No. 98301-13755
Cartridge Tape Part No. 98301-13756

Hewlett-Packard Desktop Computer Division
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 1982...First Edition

Software History

Revision 1.20 Software incorporates the following changes:

Hardware Support:
- Foreign keyboards
- Shared Resource Management via file utility
- 7470 and 7585 plotters

Graphics Editor:
- Edit system comes up in SHOW #E; mode
- Filer commands are accessible from within the Graphics Editor
- MODIFY command added
- LEVEL command added
- PLOT command enhanced
- Volume specifier changed
- EXIT available to MAIN
- Default layer number added

Printed Circuit Board Layout Module:
- VDRILL - sort option added
- VPLOT - sort option added
  arcs capability added
  circle and arcs resolution option added
- PPLOT - support output to paper tape punch added
- PDRILL - support output to 7970E magnetic tape added
  support output to HP-IB paper tape punch added
- Process File - keepout layers added
  penplot layers added
  assembly layer added
- Library Parts - four new parts

Table of Contents

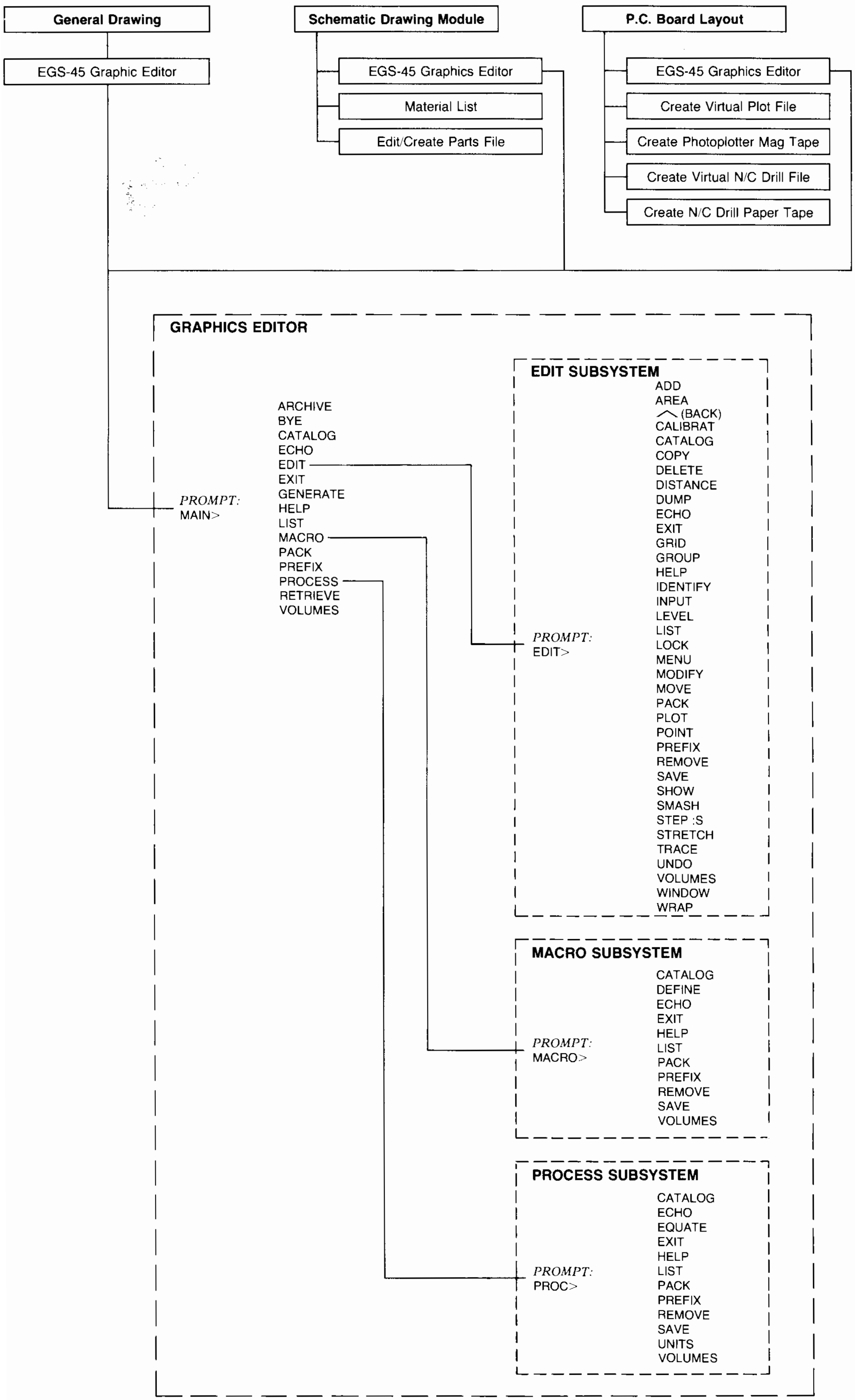
Chapter 4: System Configuration Module

Chapter 5: Command File Facility

Chapter 6: User Written Applications

Chapter 7: The Schematic Drawing Module

Chapter 8: P. C. Board Layout Module



98301-D-60212

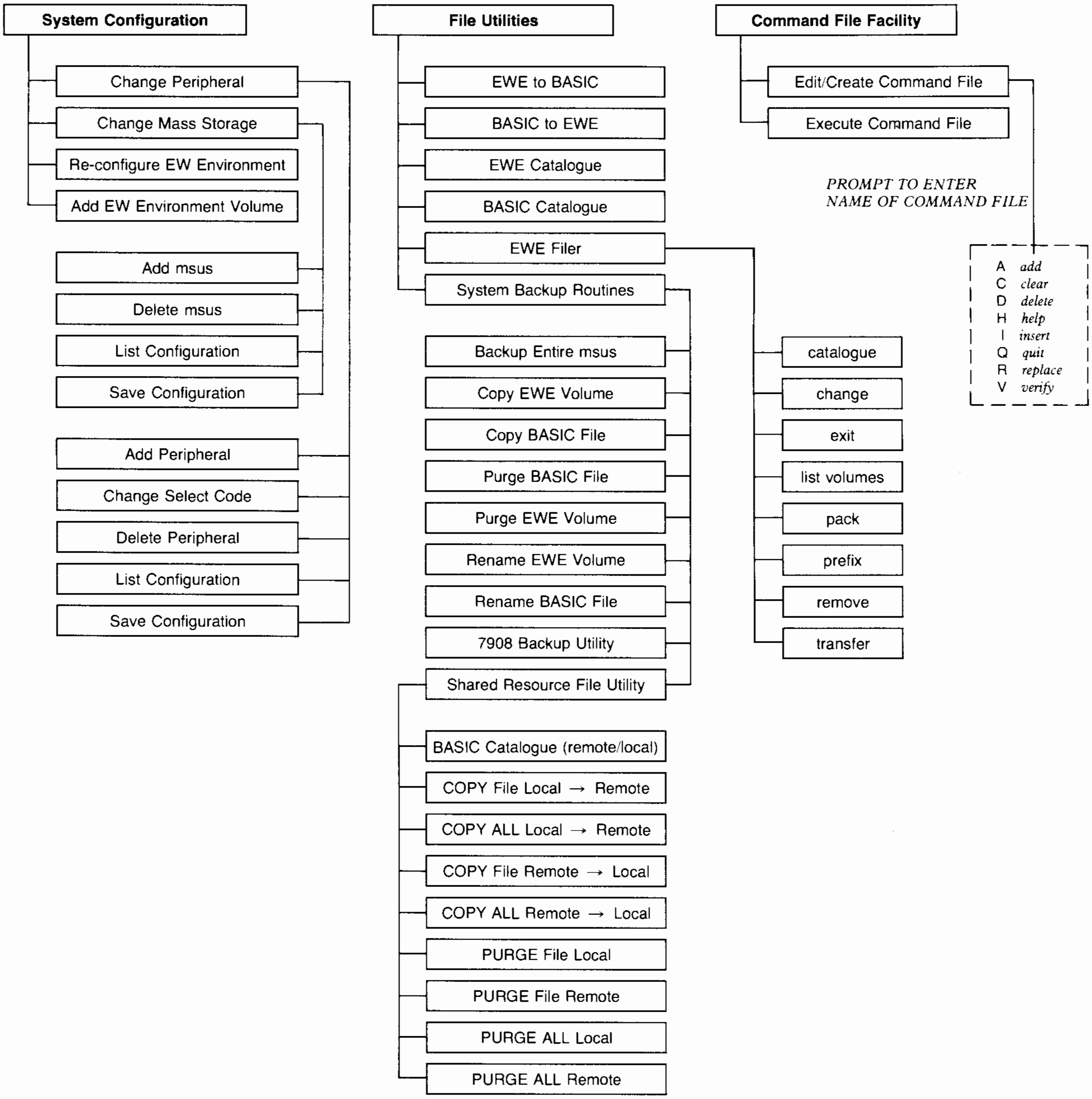

System Configuration
Change Peripheral
Change Mass Storage
Re-configure EW Environment
Add EW Environment Volume
Add msus
Delete msus
List Configuration
Save Configuration
Add Peripheral
Change Select Code
Delete Peripheral
List Configuration
Save Configuration
File Utilities
EWE to BASIC
BASIC to EWE
EWE Catalogue
BASIC Catalogue
EWE Filer
System Backup Routines
Backup Entire msus
Copy EWE Volume
Copy BASIC File
Purge BASIC File
Purge EWE Volume
Rename EWE Volume
Rename BASIC File
7908 Backup Utility
Shared Resource File Utility
BASIC Catalogue (remote/local)
COPY File Local → Remote
COPY ALL Local → Remote
COPY File Remote → Local
COPY ALL Remote → Local
PURGE File Local
PURGE File Remote
PURGE ALL Local
PURGE ALL Remote
catalogue
change
exit
list volumes
pack
prefix
remove
transfer
Command File Facility
Edit/Create Command File
Execute Command File
PROMPT TO ENTER
NAME OF COMMAND FILE
A add
C clear
D delete
H help
I insert
Q quit
R replace
V verify

Table of Contents

Chapter 1
Getting Started

Introduction

The purpose of this chapter is to acquaint you with EGS-45, show you how to set up your system, and help you create your first drawing.

First, a quick overview is provided to help you understand how EGS-45 can be used to create drawings and to give you a feel for the way in which the modules that form the EGS-45 software interact with one another. Instructions for setting up the computer and its associated peripherals are provided next, along with instructions for transferring your EGS-45 software from the cartridge tapes it is shipped on to your system's mass storage device(s). You are then shown how to load and run the EGS-45 software package. Finally, an interactive example is provided, through which you will create and store your first drawing with this system.

Overview

EGS-45 is a low cost, computer aided drafting tool which is composed of a flexible computer hardware set and the EGS-45 software package.

The EGS-45 software package is a collection of programs and data files that allow you to create, modify, store, and plot two dimensional drawings. The software is divided into the following functional groups, called modules: the General Drawing module, the System's Configuration module, the File Utilities module, the Command File module, the Printed Circuit Board Layout module (optional), and the Schematic Drawing module (optional). The diagram on the facing page shows capabilities provided by each module and shows how these capabilities are accessed. Each module (and the systems and subsystems they use) is described below. Refer to the diagram on the facing page to enhance your understanding of the discussion.

- The General Drawing module - The General Drawing module allows access to the Graphics Editor. This is its sole function and purpose.
- The Graphics Editor - The Graphics Editor is the portion of the software where drawings are created, stored, modified (sometimes referred to as editing a drawing), and plotted. The remaining modules simply provide the functions necessary to support and maintain these drawings.

  Drawings are created, stored, modified and plotted by entering commands, such as ADD, STRETCH, and PLOT, in the **edit subsystem,** within the Graphics Editor. Drawings appear on the graphic CRT as the commands are entered. Once a drawing is completed, it is stored in a special type of file, called a device file, on your computer's mass storage device.

The units of the drawing (feet, inches, millimetres,...), the line type of the drawing (or parts thereof), and the pen used to plot each part of the drawing are specified in the **process subsystem,** within the Graphics Editor. Since this information is likely to be used by many drawings, it is also stored in a special type of file, called a process file.

New commands may be created in the **macro subsystem** by combining the existing commands of the Graphics Editor and its subsystem into new commands called macros. These commands are stored in a special type of file, called a macro file. By loading a particular macro file, you may then use these macros to create, modify, store, and plot drawings.

- The File Utilities module - As you have seen, EGS-45 makes use of many special types of files which are internal to the system. This means that these files may not be accessed using BASIC language commands. Instead, you must use a set of utility programs created especially for the task. These utilities allow you to make copies of individual programs and files, make copies of collections (called volumes) of special files, make copies of BASIC program and data files, obtain listings of the special files contained in a volume, and move an individual file from one mass storage device to another or from one volume to another.

- The System Configuration module- The computer must have some method of knowing which devices are connected to it and how it can access them. This information is also stored in a special type of file, called a configuration file. The configuration file is modified and stored by functions of the System Configuration module.

  Changing the peripheral set connected to the system is referred to as reconfiguring the system. Any time that the system is reconfigured the system configuration file must be modified to reflect that change. If these changes are permanent the configuartion file will need to be saved, making the modifications to the system configuration file permanent as well.

- The Command File module - EGS-45 is interactive, which means someone must be present to enter commands one at a time to create, modify, plot, and store drawings. However, it takes some commands (such as PLOT or GENERATE) fairly long amounts of time to execute; it would be more productive to have the capacity for unattended operation of the system during these operations. The Command File module allows you to specify, at one time, the entire series of commands that you want executed. These commands are stored in a special file called a command file.

  When executing a command file, the system reads in the first command of the command file, executes it, and returns to the command file to read the next command. This continues until the contents of the command file are exhausted. Command files are most useful for post-processing drawings, such as creating the archive and generate files required by the optional application modules. For example, the Schematic Drawing module requires that an archive file be created from your schematic drawing(s) before a material list can be produced for that drawing. By creating a command file, you can create the archive files for many drawings without needing to be present to enter the commands interactively.

- The Schematic Drawing module - The Schematic Drawing module is an application module for EGS-45 system that may be purchased separately from EGS-45. It allows access to the Graphics Editor to create drawings that represent schematics. Programs supplied with the module post-process the drawings to create printed material list reports. The material list reports list the electrical parts (listed and sorted by reference designator or by your company's stock number for the electrical part) required to produce the circuit represented by the drawing.

- The P. C. Layout module - The P. C. Layout module is an application module for EGS-45 that may be purchased separately from EGS-45. It allows access to the Graphics Editor to create drawings that represent printed circuit boards. Programs supplied with the module post-process the drawings to create punched tape in Excellon format for N/C drill machines and create 1600 bpi magnetic tapes to drive a Gerber photoplotter to create the photoplots necessary to produce a printed circuit board.

The capabilities offered by these modules are accessed in one of two ways: by typing a command on the computer's keyboard or by selecting the capability from a menu on the computer's CRT.

Once a module has been selected, only the capabilities offered by that particular module may be accessed. To access the capabilities of another module you must first exit the module you are "in" (the last module selected) and then select the new module. Similarly, only the capabilities offered by a particular subsystem of the Graphics Editor may be accessed from "within" that subsystem. To access the capabiltities of another subsystem, you must first exit the subsystem you are in.

A Typical Editing Session

To help understand how these modules contribute to the overall task of creating a drawing, a list of procedures drawn from a typical editing session are provided below. Do not attempt to perform the procedures; instead use the software organizational diagram as you read to observe the use of the modules in a typical editing session.

1. Load and run the EGS-45 software.
2. If the peripherals connected to the computer have changed, select the System's Configuration module and update the configuration file to reflect the current set of peripherals connected to the computer.
3. Enter the process subsystem within the Graphics Editor to specify the units of the drawing you are to create. Additionally, specify the line type (solid, dotted, dashed, etc...) for the drawing and the plotter pen with which it will be plotted. If you plan to use the information again, save it in a process file.
4. Enter the edit subsystem and enter the commands (such as ADD, MOVE, and STRETCH) to create a drawing on the computer's CRT.
5. Since you are finished modifying the drawing, save the it in a device file with the SAVE command. This allows the drawing to be recalled later and modified.
6. Plot the drawing on the plotter attached to your computer by entering the PLOT command.
7. Exit the edit subsystem and the Graphics Editor.

8. Select the File Utilities module to obtain a listing (also called a catalogue) of the files present on the mass storage device and to make a copy (called a backup) of these files, just in case the originals are accidentally destroyed.
9. If an application module requires post-processing of one or more of the files created above, you might use the Command File Facility module.

Equipment Table

Computers

HP 9845B/C/T Computer with:

- minimum 318,000 bytes read/write memory (at least 578,000 bytes read/write memory recommended).
- at least one built-in tape drive.
- built-in 80 character printer.
- choice of one keyboard[1]
  - English
  - French
  - German
  - Spanish
  - Swedish-Finnish

ROMs

HP 98438A Assembly Language Execution ROM or HP 98439 Assembly Language Execution and Development ROM
HP 98413A Mass Storage ROM
HP 98411A Graphics ROM
HP 98412A I/O ROMs

1 Refer to Appendix A for special considerations when using EGS-45 with non-English keyboards.

Mass Storage Devices
(minimum 1.5 Mbytes using a minimum of two devices)

Equipment Description	Recommended			
	Interface	Select Code	Address	msus
HP 9885M/S Flexible Disc Drive (.5 Mbytes)	HP 98032A	8	NA	:F
HP 9895A Flexible Disc Drive ** (2.1 Mbytes)	HP 98034A/B	10	0	:H
HP 7906 Disc Drive (20 Mbytes)	HP 98041A	12	0	:C
HP 7908 Disc Drive * ** (16.4 Mbytes)	HP 98034A/B	7	0	:Q
HP 7920 Disc Drive (50 Mbytes)	HP 98041A	12	0	:P
HP 7925 Disc Drive	HP 98041A	12	0	:X
HP 9826A Opt. 500 Shared Resource Manager***	HP 98029A	5	0	:REMOTE

Graphic Input Devices

Equipment Description	Recommended		
	Interface	Select Code	Address
HP 9874 Digitizer	HP 98034A	7	6
HP 9111A/T Data Tablet	HP 98034A	7	6

Should you wish to include both a digitizer and a tablet in the system, both should be set to the same select code and address. You must exercise care that only one device is powered on at a time.

Plotters and Other Peripherals

Equipment Description	Recommended		
	Interface	Select Code	Address
HP 7225 Plotter (with HP 17601A personality module)	HP 98034A/B	7	5
HP 7470 Plotter	HP 98034A/B	7	5
HP 7580 Plotter	HP 98034A/B	7	5
HP 7585 Plotter	HP 98034A/B	7	5
HP 7970E Magnetic Tape Drive	HP 98041A	12	1
HP 9872B/C/T Plotter	HP 98034A/B	7	5
HP 9884 Paper Tape Punch	HP 98032A	2	None

* The built-in tape drive provided on this mass storage device counts as a second physical mass storage device.

** Only mass storage devices (up to two) may share this HP-IB interface.

*** The system must include a minimum of two local mass storage devices with a minimum capacity of 1.5 Mbytes in addition to the remote disc.

Setting Up the System

The Hardware

EGS-45 consists of a flexible set of computer hardware and the EGS-45 software package. The hardware required for the software consists of a computer, at least one graphic input device, a minimum of 1.5 megabytes of mass storage capacity vested in a minimum of two physical mass storage devices (other than your computer's built in cartridge tape drives), an optional, but highly recommended, plotter (such as the HP 9872B/C/T plotter), and other optional devices. The equipment table shows the supported equipment that can be used to satisfy these requirements.

Basically, setting up the system involves finding an area to place the equipment, connecting the peripherals to the computer with the interface cables, and installing the computer accessories (such as the ROMs and fuses). If you are responsible for setting up your system, please refer to the installation manual for each peripheral and the Installation, Operation, and Test manual for the computer for complete information on setting up the hardware. These manuals also provide care and cleaning information which should be read by all users.

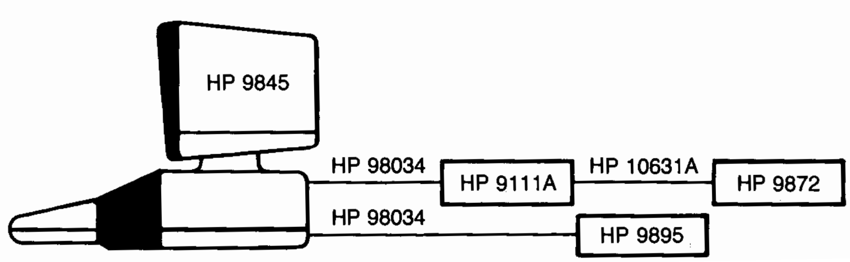
Connecting Peripherals to the Computer

Each peripheral is connected to the computer via an interface cable. Your computer's Installation, Operation, and Test manual illustrates how to connect interfaces to the computer. The equipment table shows which interface is required for each device.

HP devices such as pen plotters, printers, and graphic input devices are connected to the computer using an HP 98034A/B interface card and any of the following cables:

- HP 31389A/B/C/D
- HP 10631 A/B/C/D
- HP 10833A/B/C/D
- HP 45529A/B/C/D

The picture below shows a typical configuration. Up to 15 HP-IB devices may share one HP 98034 interface in this fashion. Mass storage devices, however, must be connected to the computer via a second HP-IB interface (if the device is an HP-IB device); mass storage devices cannot share the same interface with other devices.



Addressing Input and Output Devices

To enable the computer to "talk" to a peripheral, numeric identifiers for the interface card and the peripheral, called the select code and device address respectively, are assigned. These identifiers tell the computer where to find the peripheral just as your address tells the postman where to deliver your mail. The installation manuals for the computer and the peripherals contain information about how to set the select code and address.

The EGS-45 software expects certain select codes and device addresses for the supported peripheral devices connected to the computer (the values are preset in the configuration file when shipped). The values that are stored in the initial configuration file are indicated in the equipment table under the heading, Recommended Address and Select Code.

If you use different select codes or addresses for the mass storage device containing the system software, a plotter, or a graphic input device, the system will prompt you to indicate whether that particular device is connected to the system when the EGS-45 software is loaded. If you indicate that it is, the system then prompts you for the select code and address of the peripheral.

If you use a select code and/or address for an additional mass storage device or optional peripheral that differs from the values recommended in the equipment table, it is necessary to modify the configuration file via the System Configuration module before the peripheral may be accessed.

Turning On and Testing the System

Once all peripherals are connected to the computer and their power cords plugged in, turn on each piece of equipment. The CRT displays MEMORY TEST IN PROGRESS and, after a few seconds, 9845 READY FOR USE.

Once the system is set up and turned on, you can verify that it is operating properly by running various tests. These tests are detailed in the computer's Installation, Operation, and Test manual.

The Software

Shipped with EGS-45, besides the computer and peripherals, are the following items:

- 3 tape cartridges containing the EGS-45 software (HP part numbers 98301-13754, 98301-13755, and 98301-13756).
- this manual: EGS-45 Operating Manual (HP part number 98301-12111).
- document binder, (HP part number 9282-0895).

If you ordered one or more of the optional application modules with EGS-45, items for that module may be included with this shipment. For information about the application modules, please refer to the manual section for that module.

Please check to insure that you received all the items you ordered. If any items are missing, please contact you local HP Sales and Service office.

Preparing the Software for Use

Before the software can be used it must be transferred to the disc drive you intend to use with this system. A program named COPY has been provided on tape number 1 (General Drawing Core #1) for this purpose. To use this program, follow the steps below.

1. Turn on your computer and your mass storage device(s).
2. Make sure that the mass storage medium is initialized. For instructions on initializing a mass storage medium refer to your computer's programming manual.

   When initializing HP 9885 discs, use the interleave factor, 7.

   When initializing HP 9895 discs, use the interleave factor, 11.
3. Make sure that there is paper in the computer's built-in printer.
4. Insert EGS-45 software tape 1, General Drawing Core #1 (HP part number 98301-13754) in the right-hand tape drive on the computer.
5. Enter the following command on your computer:

   `LOAD "COPY:T15",1` (EXECUTE).
6. The computer prompts you for the msus (Mass Storage Unit Specifier: see the computer's Mass Storage ROM manual for information) of the device containing the software to be copied, called the source device. You should respond be entering:

   `:T15` (CONTINUE).
7. Next the system prompts you to enter the mass storage unit specifier that identifies the mass storage device to which you want to copy the software, called the destination device. If you used the default select code and address for your mass storage device, refer to the msus provided in the equipment table. Otherwise, refer to your computer's Mass Storage manual for details. Enter the appropriate msus and press the CONTINUE key.

   The system then displays the msus entered for the source device and the destination device, and asks you if these are correct. Entering N, for no, causes you to return to step 6 above, thus allowing you to enter new source and destination msus.
8. Press (C) to initiate the software transfer of all files on the source device to the destination device. The computer will display `COPY COMPLETE` when all files are copied*.
9. To copy the software from EGS-45 software tape 2, repeat procedures 6 through 8 after first inserting tape 2 and pressing the CONTINUE key (General Drawing Core #2 ; HP part number 98301-13755).

   To copy the software from EGS-45 software tape 3, repeat procedures 6 through 8 after first inserting tape 3 (General Drawing Core #3; HP part number 98301-13756) and pressing the CONTINUE key.

* Should the copy program encounter an existing file with the same name as the file being copied (as might occur when copying a new software release to the disc), you have three options.
1. Purge the existing file before making the copy or do not copy.
2. Automatically purge all existing files before performing the copy.
3. Skip duplicate files and continue with the next file.

The options are selected by pressing C, P or S, as instructed by the program.

You may now use the software that you have copied to your mass storage device. Store the original tapes (Drawing Core #1, #2, and #3) in a safe place in case the copy on your disc is accidentally destroyed. You may wish to keep additional copies of the software on disc to serve as a backup to the disc you just created, again in case the disc is destroyed. You will learn more about backups in Chapter 4, with the File Utilities module.

Loading the Software

To load and run the EGS-45 software, execute the following steps:

1. Return your computer to its initial state by entering

   ```
   SCRATCH A (EXECUTE)
   ```


2. Next, enter

   ```
   LOAD "AUTOST:msus",1 (EXECUTE)
   ```

   where msus stands for the mass storage unit specifier of the mass storage device containing the software: for more information see your computer's Mass Storage ROM manual .
3. The system prompts you to enter the msus of the device containing the EGS-45 software. Enter the appropriate msus and press the CONTINUE key.
4. When the system prompts you for the current date, enter the date (for example, enter 08-OCT-81 for October 8, 1981), by using the left/right arrow keys located at the top of the computer keyboard, the insert and delete character keys, and the alphanumeric keys available on the keyboard and numeric keypad. Then press the CONTINUE key.
5. The system next checks to see if all of the devices listed in its configuration file are present, and prompts you if a graphic input device or plotter is not found. If you respond that the devices are connected to the computer, the system then prompts you for the select code and device address of the peripheral(s).

   If you have other peripheral devices connected to the computer (such as a printer, another mass storage device, a paper tape punch, etc.) and have not used the suggested default select codes and addresses, you will need to modify the configuration file to reflect the values that you used. These changes do not need to be made until you are ready to use the device. Information on changing the configuration file is found in chapter 4 of this manual.
6. The system then displays tables of all mass storage devices, all system peripherals, and all volumes that are found on the system. A volume is a collection of special files (such as device files, process files, ...) used by EGS-45 or is a physical device (such as the computer keyboard, or the system CRT display). The concept of volumes is discussed in more detail in Chapters 2 and 3 of this manual.

SYSTEM PERIPHERALS	TYPE	SELECT	ADDRESS
#1	9111A	7	6
#2	9872T	7	5
#3	PRINTER	0	

SYSTEM MASS STORAGE DEVICES	MSUS	DEVICE NAME
# 1	:F8	9885 FLOPPY DISC
# 2	:F8,1	9885 FLOPPY DISC

EDITOR WORKSTATION ENVIRONMENT VOLUMES		
VOLUME #	VOLUME LOCATION	VOLUME NAME
# 1	KEYBOARD	
# 2	CONSOLE	
# 3	:F8	EWSYS
# 4		
# 5		
# 6	0	PRINT

The system next displays a menu of the different modules that may be selected. By comparing this menu with the software organizational diagram located at the beginning of this chapter, you can see that along with each module displayed in the menu is the first set of capabilities of that module. Use the ROLL up and ROLL down keys located at the top center of your computer keyboard to view and select the optional application modules, if they are present. You may either select a module, and then select a capability, or select the capability directly without first selecting the module.

This menu is often referred to as the **Manager** or **System Manager.**

The order of the modules in the display may vary if the software is not copied from the supplied tapes to your mass storage as instructed.

- **General Drawing**
  - EGS-45 Graphic Editor

- **Schematic Drawing Module**
  - EGS-45 Graphics Editor
  - Material List
  - Edit/Create Parts File

- **P.C. Board Layout**
  - EGS-45 Graphics Editor
  - Create Virtual Plot File
  - Create Photoplotter Mag Tape
  - Create Virtual N/C Drill File
  - Create N/C Drill Paper Tape

- **System Configuration**
  - Change Peripheral
  - Change Mass Storage
  - Re-configure EW Environment
  - Add EW Environment Volume

- **File Utilities**
  - EWE to BASIC
  - BASIC to EWE
  - EWE Catalogue
  - BASIC Catalogue
  - EWE Filer
  - System Backup Routines
    - 7908 Backup Utility *
    - Shared Resource File Utility *

- **Command File Facility**
  - Edit/Create Command File
  - Execute Command File

* Note: These modules are only displayed if the appropriate hardware and software are installed in your system.

The Autostart Feature

Your computer has a feature called autostart, which causes a program to be automatically loaded and run by simply switching on the power to the computer. This feature requires that the program is stored on a cartridge tape with a file name, AUTOST. The tape must be placed in the right-hand tape drive of your computer and the key labeled, AUTOST, must be latched in the down position. When power is switched on, the computer automatically loads the file, AUTOST, from the tape and runs it.

You may take advantage of the autostart feature of your HP 9845 computer by making a copy of the program, AUTOST, on an initialized, blank, cartridge tape. To do this, initialize a blank cartridge tape, insert the initialized tape into the right hand tape drive of your computer, insert your EGS-45 software disc in your mass storage device, and type the following on the computer keyboard:

```
COPY "AUTOST:msus" TO "AUTOST:T15" (EXECUTE).
```

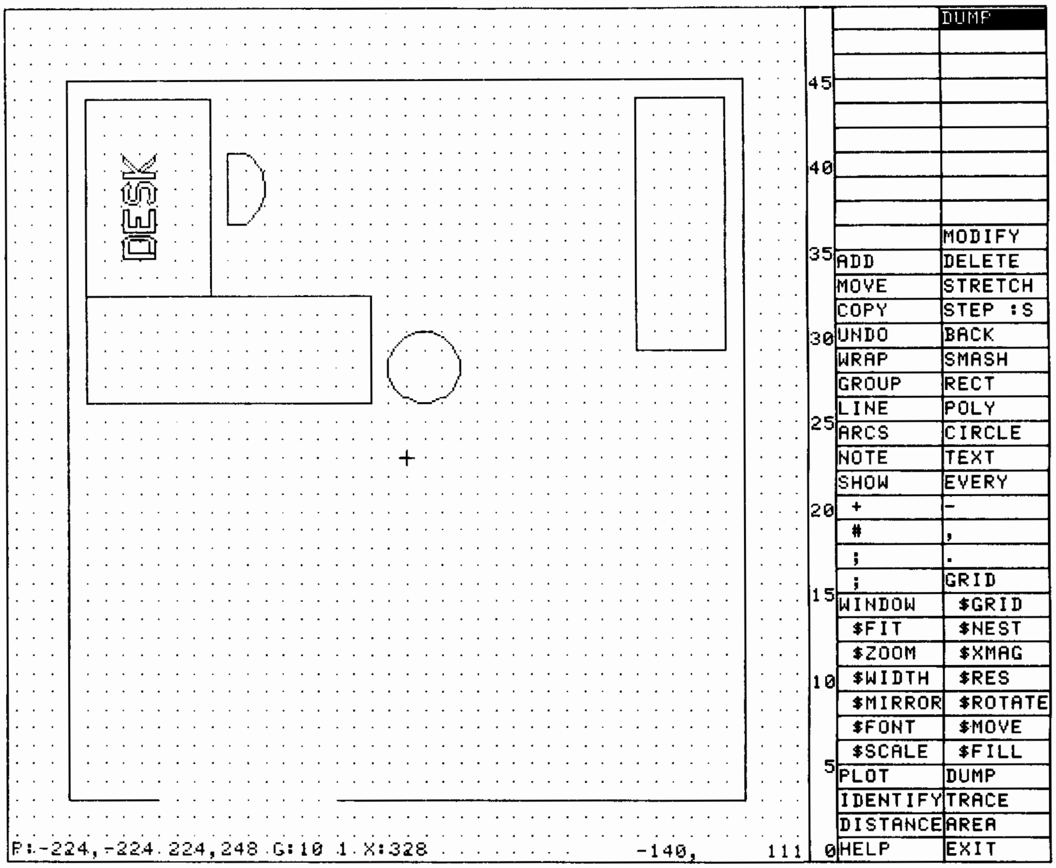
where msus is the msus of the mass storage device containing your system software.

By keeping this tape in the right-hand tape drive and latching the AUTOST key on the 9845's keyboard, you can avoid steps 1 and 2 in the loading process described above by simply turning the computer off (for about 10 seconds) and then on again. This causes the program, AUTOST to be loaded from the tape and run. For more information, see AUTOSTART in your computer's operating manual.

A Simple Drawing

Before going on to study the details of the software package, a simple interactive example is presented. By working along with this example, you can get a better understanding of the way the system works. Refer to the system diagram at the beginning of this chapter to help you understand the way that the different modules and their capabilities are utilized.

The drawing that you create in this example is shown below.



To begin the example, you must first learn to use your graphic input device (either the HP 9111 or the HP 9874). Place the stylus (or optional puck, if you are using the 9874) on the platen of the graphic input device. A small crosshair should appear on the computer's CRT. Moving the stylus across the platen causes the crosshair to track your movements on the CRT. To select a module, move the stylus until the crosshair is located directly over the module name, and press the point of the stylus into the platen until a click is heard (or if you are using the optional puck with the 9874, press the button on the puck labeled D until a click is heard). This is called selecting, or digitizing.

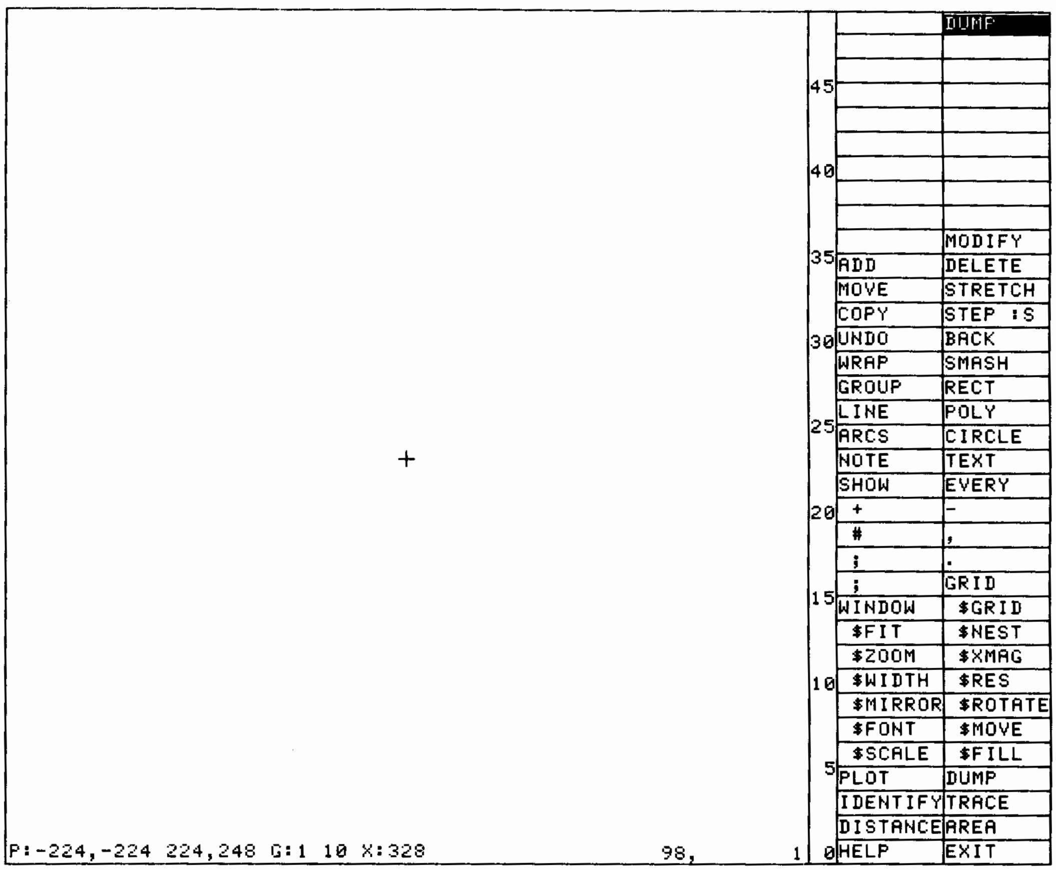
To create the example drawing, perform the following steps.

1. Load and run the EGS-45 software.
2. Select the Graphics Editor from the menu on the CRT. Use the method described above or use the left-right and up-down arrow keys located at the top center of the keyboard to cause the Graphics Editor to be highlighted (then press the CONTINUE key).

On systems equipped with a standard top (systems ordered with Opts. 1XX), selection of modules with the graphic input device is not possible. You must use the left-right and up-down arrow keys to select a module.

3. Enter the edit subsystem by typing

   `EDIT ;` (CONTINUE). The CRT should now match the display pictured below:



You will use the technique of selecting to enter commands from the menu area of the CRT and to specify x,y locations in the drawing area of the CRT. The numbers in the lower right hand corner of the CRT indicate the x,y location of the cursor, in current units, when digitizing points (selecting locations in the drawing area). Verify this by moving the stylus of your graphic input device to cause the cursor on the CRT to move across the drawing area and observe how the location values change.

Numeric values often required with commands may be selected from the numeric column in the display. For example, to enter the number 90, position the cursor over the value 9 in the numeric column and select. In a similar fashion, select the value 0 to complete the entry. In this manner values larger than 45 may be entered.

The top right slot in the menu displays the last command entered in the edit subsystem, called the current active command.

The drawing area of the CRT can be thought of as a stack of 255 sheets of clear plastic, called layers. Drawings may be composed of images in one layer, or in many layers. Layers may be displayed individually or in any combination with others. In this manner, portions of the drawing may be turned off (not displayed) to help clarify the drawing without losing the information. This concept is discussed fully with the edit subsystem concepts in Chapter 2.

4. Enter the following commands (select the individual items from the menu area and the numeric evaluator column area of the CRT). If you make a mistake entering one of these commands, simply select the semicolon (;) from the menu (terminating the previous entry) and enter the command again.

```
GRID 10,1 ;

GRID;
```

   This causes a grid to be displayed in the drawing area of the CRT. Each grid point represents 10 units. When specifying x,y locations in the drawing area, the point selected snaps (shifts) to the nearest grid point displayed. Thus, it is only necessary to be close to the locations specified, not exact.

5. If an error is made when entering a command during this example, it is not catastrophic. Three ways of recovering from an error are discussed below.

   a. If an error is made while specifying x,y locations, try selecting the BACK command. For example, enter the following series of commands and observe the results on the CRT (the paired numbers separated by a comma are x,y locations to be selected from the drawing area):

```
ADD LINE 1 -100,-100 100,-100 100,100

BACK

BACK

BACK
```

   The message "incomplete line" is displayed in the top left portion of the CRT, indicating that all x,y points entered with the ADD command have been deleted. You may ignore this message.

b. If the command has already been terminated by entering a semi-colon, try selecting the UNDO command. For example, enter the following series of commands and observe the results on the CRT:

```
ADD RECT 1 -100,-100 100,100 ;

UNDO
```

c. If the command has been terminated by entering a semicolon, use the DELETE Command.

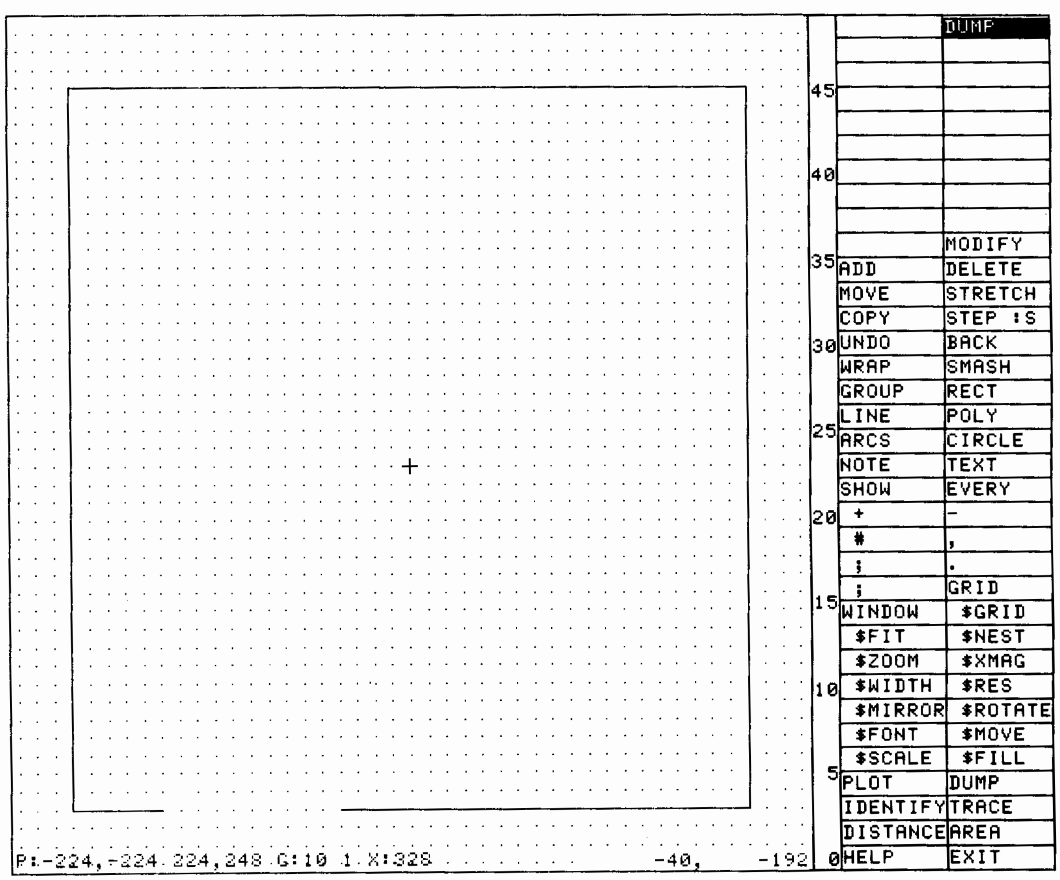
For example, enter the following commands and observe the results on the CRT:

```
ADD RECT 1 -100,-100 100,100 ;

DELETE 0,0;
```

6. Create the room area of the floor plan drawing by entering the following command:

```
ADD LINE 1 -140,-190 -190,-190 -190,210 190,210 190,-190
-40,-190 ;
```

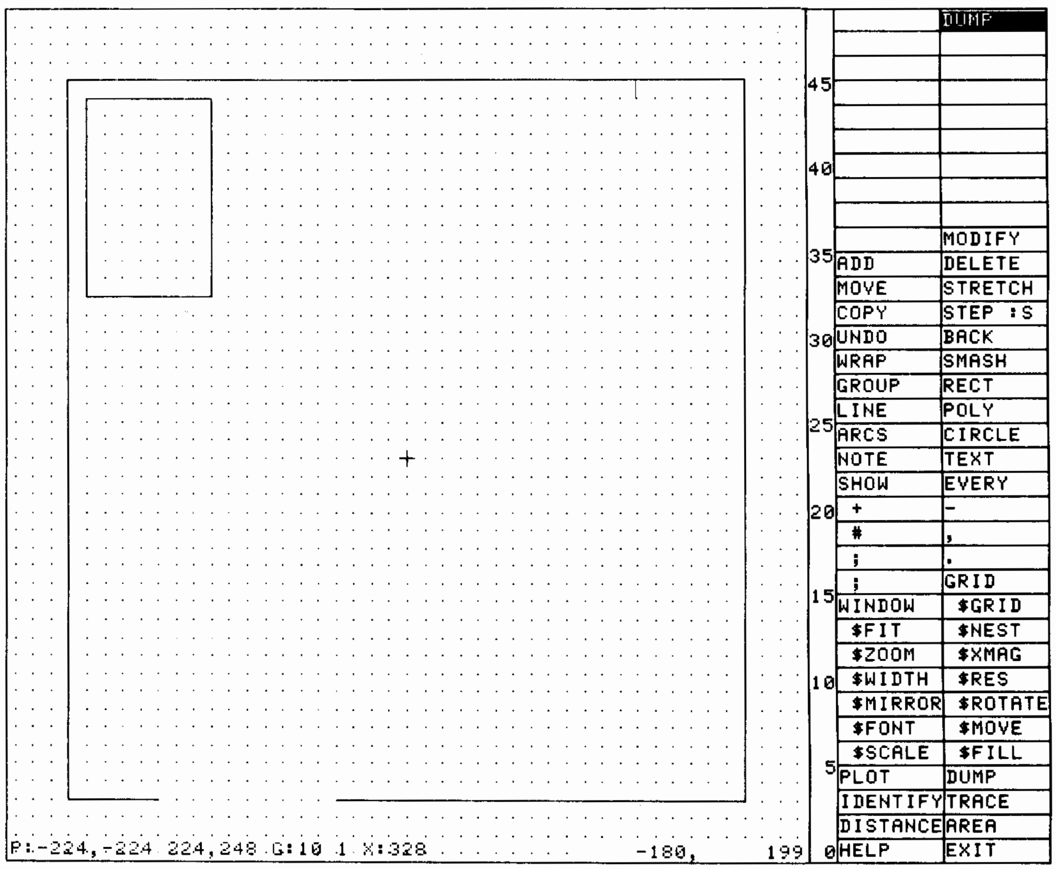
This causes a line to be plotted on layer one of the CRT.



7. Add a desk to the floor plan by entering the following command:

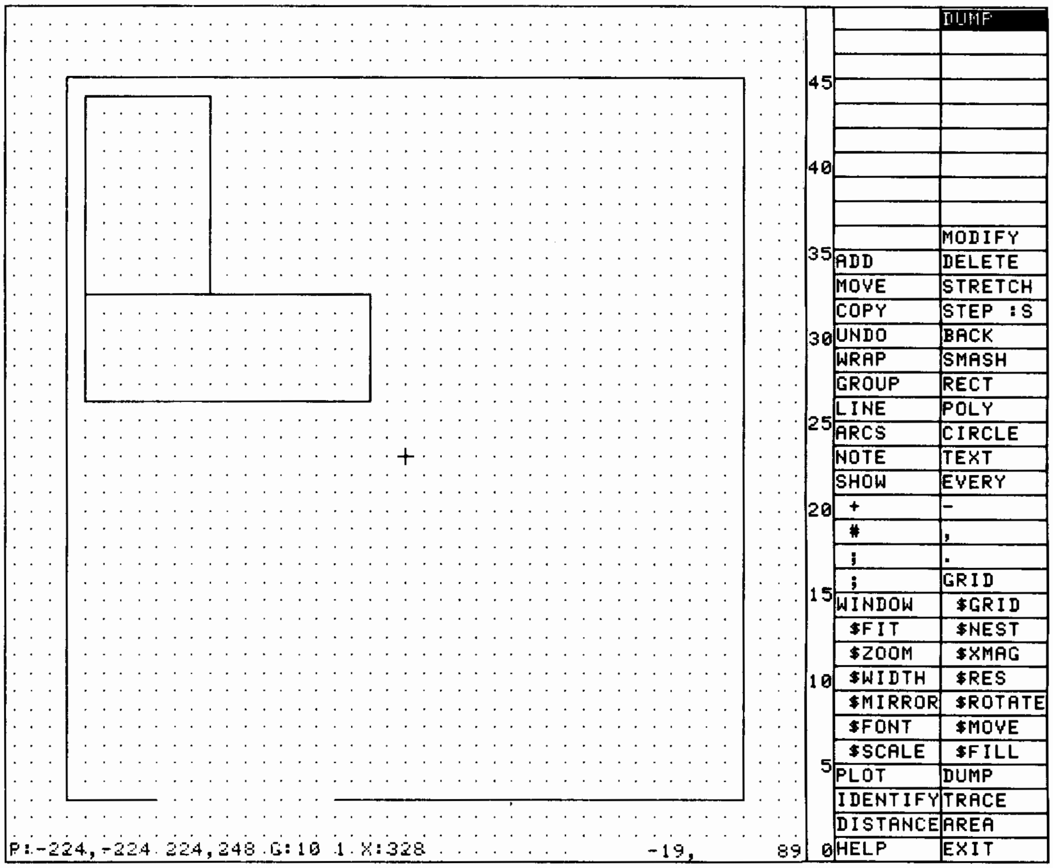
```
ADD RECT 2 -180,90 -110,200;
```

This adds a rectangle to layer two of the drawing. By placing the room outline on layer one and the furniture for the floor plan on layer two, the drawing can be viewed with just furniture, just the room outline, or with both.



8. Add a wing to the desk in your floor plan by entering the following command:

```
ADD RECT 2 -180,30 -20,90;
```

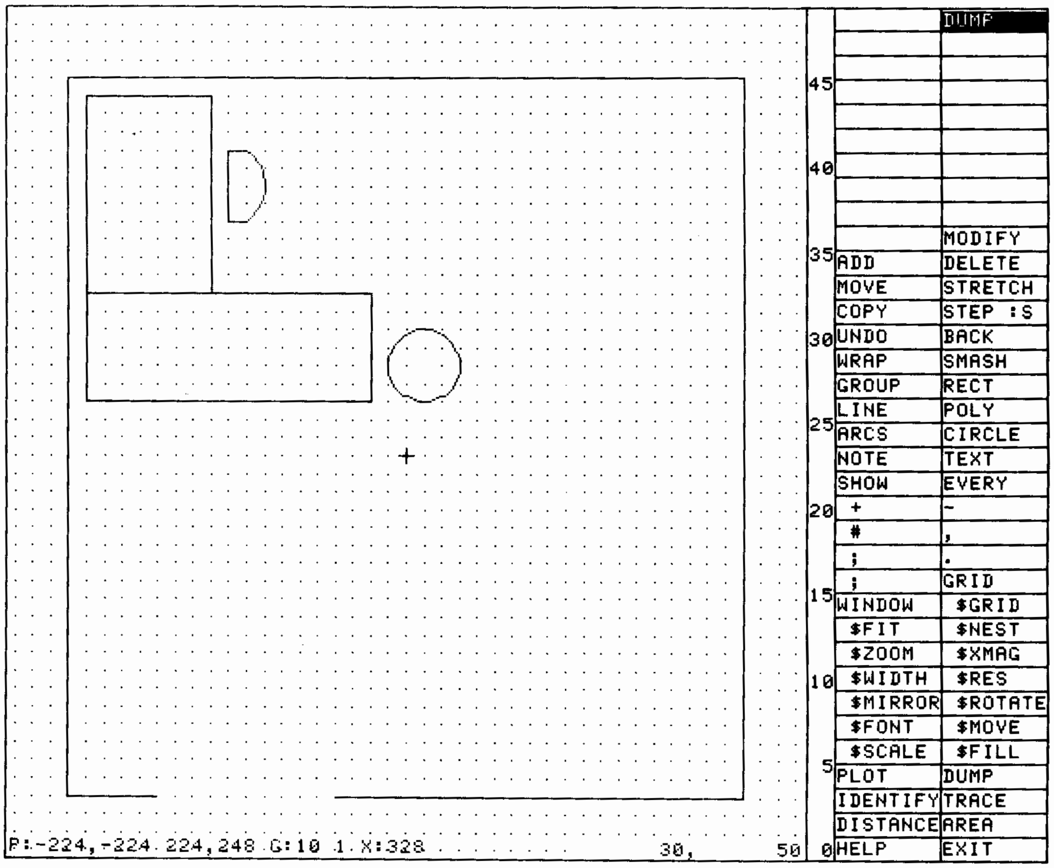


9. To create the chair for the floor plan, enter the following two commands:

```
ADD LINE 2 -90,170 -100,170 -100,130 -90,130 ;

ADD ARCS 2 $RES 6 -90,130 -90,170 -80,150 ;
```

10. Since no desk is complete without a wastebasket, enter the following command:

```
ADD CIRCLE 2 $RES 6 10,50 30,50 ;
```

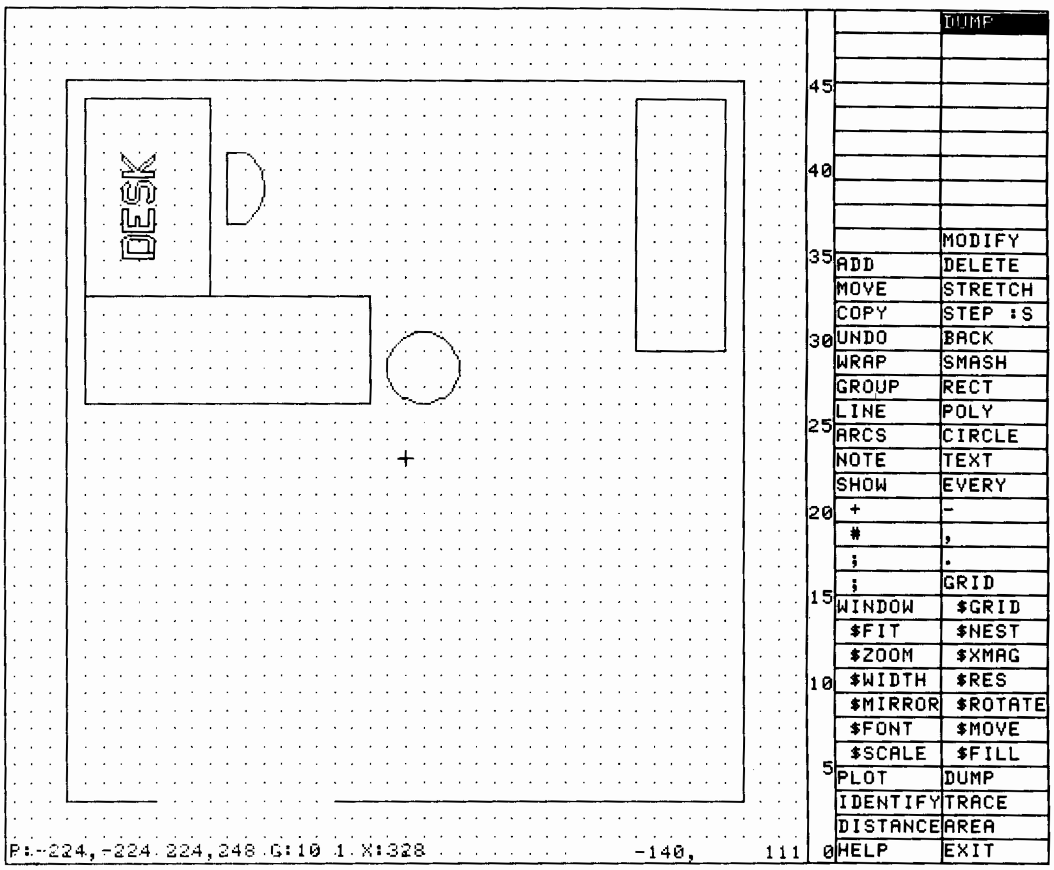


11. Add a bookcase to the floor plan by entering the following command:

```
ADD RECT 2 130,60 180,200 ;
```

12. Add a textual label to your drawing that identifies the desk by entering the following command. You will need to enter the characters "DESK" on your computer's keyboard (then press CONTINUE):

```
ADD TEXT 3 $ROTATE 90 $FONT 20 'DESK' -140,110 ;
```



13. To save the drawing you just created so that you may use it again in later examples, enter the following command by typing it on the computer keyboard:

    ```
    SAVE EXAMP ;
    ```

    and press the CONTINUE key. This saves the drawing in a device file with the name, EXAMP.

14. To get a copy of the drawing you just created, enter:

    ```
    DUMP
    ```

15. Exit the edit subsystem by entering:

    ```
    EXIT ;
    ```

16. Exit the Graphics Editor by entering:

    ```
    BYE ;
    ```

    and press the CONTINUE key.

You have just completed working an interactive example that provided your first experience with drawing creation with EGS-45. The remainder of this manual elaborates on the capabilities provided with each module and the concepts necessary to make use of these capabilities. You will make use of the drawing that you created above in other chapters as part of a continuing interactive example that teaches you how to use the system.

Use the system overview provided at the beginning of this chapter along with the software organizational diagram as you study each module and its capabilities in the remaining chapters.

Table of Contents

Chapter 2
The Graphics Editor

Introduction

The purpose of this chapter is to familiarize you with the Graphics Editor and its subsystems.

First, an overview is provided to help you understand the function of the Graphics Editor and the way it interacts with the rest of the EGS-45 software. Next some basic concepts are discussed, such as a brief introduction to files and volumes, commands, and the functions of the subsystems. The commands and capabilities provided by the Graphics Editor are discussed.

A similar presentation is then provided for each of the Graphics Editor subsystems (the edit, macro, and process subsystems): an overview, some basic concepts, and finally the commands and subsystem capabilities.

Overview

The Graphics Editor is the portion of the EGS-45 software where drawings are created, stored, modified (edited), and plotted. Most of these capabilities are provided through one of the three subsystems contained in the Graphics Editor (refer to the foldout system software diagram at the beginning of this manual).

The Graphics Editor itself provides a limited number of commands which are entered in response to the prompt:

```
MAIN>
```

When this prompt is displayed on the CRT, it serves as a landmark indicating that you are "in" the Graphics Editor.

Three of these commands allow access to the subsystems of the Graphics Editor where many more commands are available. The remainder of the commands in the Graphics Editor provide the tools for analyzing files containing drawings (such as the GENERATE and ARCHIVE commands) and for providing general information about commands or use of the system (such as HELP or ECHO).

The edit subsystem, contained within the Graphics Editor, provides the commands necessary to create, modify, plot, and store a drawing. Drawings are stored in special files, called device files, which may only be accessed with the EGS-45 software.

The process subsystem, also contained within the Graphics Editor, allows you to determine the units (such as feet, inches, centimetres, etc.) of the drawing, the line type used to represent portions of the drawing, and other parameters which define how the drawing is to be displayed in the drawing area of the CRT. This information may be stored in a special file, called a process file, which may only be accessed with the EGS-45 software.

The **macro subsystem** allows you to create new Graphics Editor commands or commands for any of the subsystems by combining one or more existing commands. These macro definitions may be stored in a special file, called a macro file, which, like device files and process files, may only be accessed with the EGS-45 software.

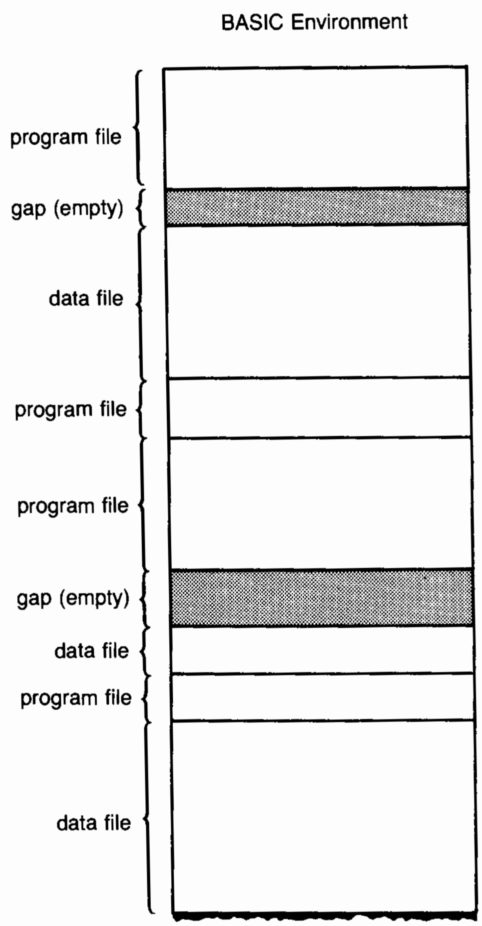
Concepts

Files, Volumes, and Environments

Files and volumes are briefly introduced in this chapter to help you understand how to use the files required and created by the Graphics Editor. A more thorough discussion of these subjects is found in the File Utilities chapter.

To begin, let us consider a BASIC program or data file. This is a collection of computer readable program lines or data that is stored on your mass storage media. It is called a BASIC file because it may be read or accessed with BASIC langauge commands and statements (BASIC is the program language of the HP 9845 Computer). Your computer's programming manual provides an in-depth discussion of BASIC files.

Physically, the computer stores programs and data in a rather complex manner on the mass storage device. However, for conceptual purposes, BASIC program and data files stored on a disc may be thought of in terms of the drawing below. Here a file is shown as a continuous block of information. Files may be contiguous or gaps may exist between files on the disc. The size of the file is measured by the amount of area on the mass storage media it occupies.



Since these files may be accessed with BASIC language commands, we say that these programs and data files are in the **BASIC environment.**

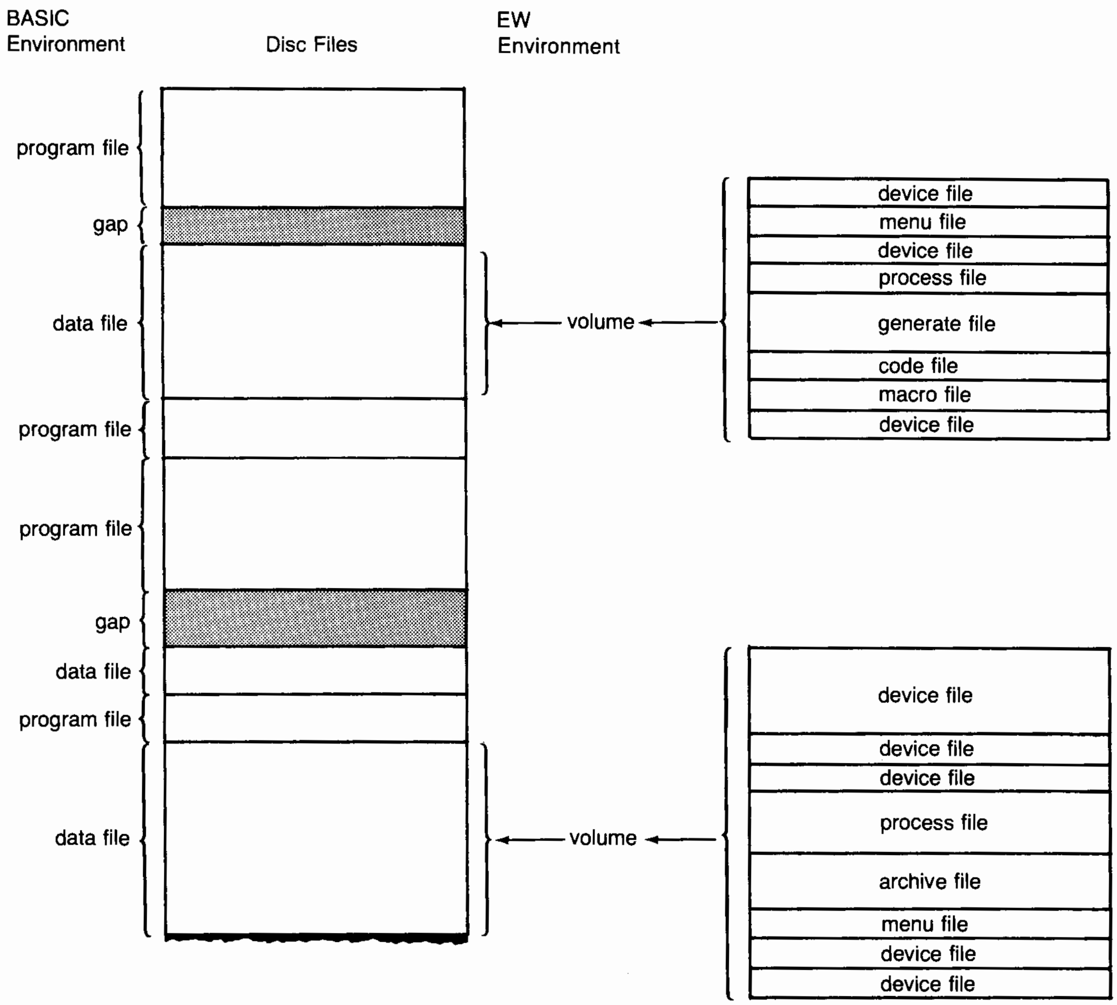
In contrast, the files created and used by the EGS-45 software are said to be in the **Editor Workstation Environment (EWE)** . Examples of EWE files are:

- Device files which contain drawings
- Process files which define the drawing area of the CRT
- Macro files which contain new commands that you create.

EWE files are accessible only through the EGS-45 software.

Many EWE files are stored together in an EWE volume, which is nothing more than a large, single BASIC data file. The volume is addressed by its volume name (which is equivalent to the BASIC data file name) or by a volume number assigned by the system.

The drawing below shows how the files of a single disc are viewed in both the BASIC environment and in the EW environment.



You have already created one type of EWE file. When you saved the example drawing EXAMP that you created in Chapter 1, the drawing was stored in an EWE device file. A device file is an EWE file in which drawings (also called devices) are stored. Other types of EWE files include process files, macro files, generate files, and archive files. Discussion of these files is presented with the commands that create the files and with the concepts section of Chapter 3.

In summary, an EWE file is a special file used by the EGS-45 software. It is stored in an EWE volume which is actually a BASIC data file. EGS-45 provides the tools to access these EWE files, which cannot be accessed with BASIC language commands or program statements.

Auto-load Files

When the Graphics Editor is selected, several EWE files are automatically loaded into the system. The name and function of each file is described below:

- MACRDATA - this macro file contains a set of macro commands. Macro commands are commands created by combining one or more existing commands into a new command,

and by assigning the new command a (macro) name. The set of macros supplied with the system are designed to replace some of the more cryptic command/parameter combinations used in the edit subsystem. A complete list and explanation of these macros is provided with the DEFINE command in the macro subsystem section in this chapter.

- MENUDATA - this menu file defines the items that appear in the menu portion of the display in the edit subsystem. This typically consists of a number of often used edit subsystem commands, macros, and library part names. The menu provides an alternate method of command entry. Commands may be entered by selecting the command name and/or any parameters from this menu area. The contents of the file MENUDATA is detailed with the MENU command in the edit subsystem section of this chapter.
- PROCDATA - this process file specifies the units (feet, inches, centimetres, etc) in which a drawing is to be represented. Additionally, it defines each of the layers upon which the drawing is created (this concept is discussed with the edit subsystem concepts). Defining a layer involves, among other things, specifying which line type (solid line, dashed line, etc) the portions of the drawing in that layer are to be displayed. The information contained in the process file PROCDATA, as furnished, is discussed with the EQUATE command in the process subsystem section of this chapter.

By using the commands of the edit, macro, and process subsystems you can modify the contents of these files to reflect the needs of your application. By then storing these modified files with the appropriate name (MACRDATA, MENUDATA, or PROCDATA) you can cause your customized file to be automatically loaded into the system when the Graphics Editor is selected.

Command Entry

Commands are key words (and parameters) which tell the computer what you want to do. EGS-45 has a vocabulary of commands that is illustrated in the foldout system software diagram, at the beginning of this manual.

Commands in the Graphics Editor are entered by typing the command name and any parameters on the computer keyboard. Only the first three characters of a command name need be entered (for example, entering PRO is equivalent to entering the entire command name, PROCESS). Additional characters of the command name may be entered; however, they must be the correct characters in the correct order. For example, GEN, GENE, GENER, and GENERAT are all correct entries for the GENERATE command. However, GENERT is not a correct entry for the GENERATE command because the character T should have the character A before it.

Most commands of the Graphics Editor and its subsystem are terminated with a semicolon. Additionally, entering another command terminates the previous command, if no semicolon was entered with it.

Pressing either the CONTINUE key or the STORE key tells the system that you are finished typing the command. Pressing the EXECUTE key has the same effect as pressing either STORE or CONTINUE; however, a semicolon (;) is automatically appended to the end of the characters entered. For example, typing EDIT and pressing the EXECUTE key causes `EDIT;` to be entered. Since many of the commands in the Graphics Editor and its subsystems require a semicolon at the end of the command, this is a handy feature.

Graphics Editor Commands

In this section, each of the commands available in the Graphics Editor is described. Since many of these commands operate on files created in the edit subsystem, you may find it helpful to read this section, then read the section of this chapter dealing with the edit subsystem, and finally return and read this section again. This will help you understand some of the terms and concepts, such as devices and drawing hierarchy.

The discussion of the Graphics Editor commands is limited to a functional description and an explanation of the options provided. Exact command syntax is provided in the Command Reference section of this manual, along with several examples of the use of each command.

ARCHIVE command

The ARCHIVE command allows you to convert a device file (a file containing a drawing) into an archive file. An archive file is an EWE file that contains a list of the commands of the Graphics Editor and its subsystems used to create a drawing. If the drawing contains links to other devices, these links are included in the archive file.

The information contained in the archive file is in human readable, ASCII format. By using the file utilities or the programming tools provided with EGS-45, your BASIC language application programs may access this list of commands, allowing you to post-process your drawing. The programming tools provided with EGS-45 are discussed in Chapter 6 of this manual.

The ARCHIVE command also allows you to obtain a listing of the archive file on the current system printer, while the file is being created.

Archive files are often used by the optional application modules available with EGS-45.

BYE command

The BYE command allows you to exit the Graphics Editor. The manager menu is again displayed, thus allowing you to select another module. A warning message is displayed if any devices (drawings) have been created in the edit subsystem but not saved on your mass storage media. The warning message lists these device names and states that these devices will be lost if the Graphics Editor is terminated before the devices are saved. You may either re-enter the edit subsystem and save the device(s) or exit the Graphics Editor by entering the BYE command again. Any devices, process files, and macro files created during the edit session that have not been saved are lost when you exit the Graphics Editor.

CATALOG command

The CATALOG command will allow you to obtain a catalog list of any volume while in the graphics editor. The resulting file list will be output to the current system printer. This command is useful in that it allows you to look into a volume at any time from any subsystem, to find a particular file name.

ECHO command

The ECHO command allows you to enable/disable echo. When echo is enabled, each command is printed on the current system printer as it is entered. This provides a log of the commands entered which makes it easy to find an incorrect entry.

When echo is disabled, commands are displayed on the CRT as they are entered, but are not copied to the printer.

EDIT command

The EDIT command allows you to enter the edit subsystem to create a drawing from scratch or to modify an existing drawing. Once in the edit subsystem, a large number of commands are available which allow you to add components to the CRT drawing area, modify these components, plot the drawing, and store the drawing in a device file.

Entering a device name with the EDIT command allows you to edit (modify) the device which is stored in that device file. The device is loaded into computer memory and is displayed on the CRT as you enter the edit subsystem.

If no device file exists with the supplied device name or if no device name is supplied with the EDIT command, no device file is loaded as you enter the edit subsystem. This allows you to create a new device (drawing).

GENERATE command

The GENERATE command allows you to convert a device file into a generate file. A generate file, like an archive file, is an EWE file that contains a list of the commands of the Graphics Editor and its subsystems used to create a drawing. If the drawing contains links to other devices, the links are NOT included in the generate file. Instead, they are replaced with the components of the devices to which they link.

The information contained in the generate file is presented in ASCII format for your use. By using the file utilities or the programming tools provided with the system, your BASIC language application programs may access this list of commands, allowing you to post-process your drawing.

The GENERATE command also allows you to obtain a listing of the generate file on the current system printer, while the file is being created.

Generate files are often used by the optional application modules available with EGS-45.

HELP command

The HELP command allows you to obtain a quick syntax reference for any or all commands valid in the Graphics Editor or one of its subsystems. For example, the HELP command entered in the macro subsystem allows you to obtain a quick syntax reference for any single command or all commands valid in the macro subsystem.

The command for which you seek help must be a valid command in the section of the software (i.e., the Graphics Editor or a subsystem) from which HELP is entered. For example, entering HELP ARCHIVE from inside the edit subsystem is not allowed, since the ARCHIVE command is not a valid command in the edit subsystem.

When you enter the HELP command, the syntax of the specified command is displayed on the CRT. When no command is specified with HELP, all syntax for all valid commands are printed on the current system printer. To explain the syntax representation, consider the following:

```
HELP ARCHIVE ;

ARChive [(+|-|#)desc[layer]] [:L] [:Nlvl] [volume] name [volume] EOC
```

The command name is displayed showing the first three letters of the command name in capital letters with the remainder of the command shown in lower case. This indicates that only the first three letters of the command are required when entering the command name.

Items enclosed in square brackets [ ] represent optional parameters that may be entered with the command. Items represented in lower case within the brackets (such as layer, cmpnt, or vol) are short notes indicating the type of data or parameter expected. Colons (:) followed by a capital letter are characters indicating an option and must be entered exactly (for example, :L or :N).

Items enclosed in braces { } represent a choice of items, where at least one of the choices must be entered. Choices are separated by a bar (|). For example, in the command above, {+ | - | # }indicates that either the plus sign (+), the minus sign (-), or the pound sign (#) must be entered.

EOC is a mnemonic for "end of command" and represents a command terminator. The command terminator is the semicolon (;) character. If the command requires xy location data with the command (such as the ADD or the COPY commands of the edit subsystem), repeating the last xy point entered also acts as a command terminator (EOC). Entering a new command before terminating the previous command automatically terminates the previous command.

These help messages are intended to be a quick reference for the experienced user. The help message for each command is listed with the syntax diagrams in the syntax reference chapter of this manual. In this way you can quickly learn to associate the short parameter descriptions with the longer explanations provided with the syntax diagrams.

LIST command

The LIST command allows you to get a listing of the current file(s) (macro and process) and device(s) loaded into the system. Additionally, you can obtain a listing of the contents of the current macro or process files. A listing of a device provides, among other things, a list of the components and devices utilized by the device. The listings are printed on the current system printer, as defined in the configuration file.

MACRO command

The MACRO command allows you to load a particular macro file and/or enter the macro subsystem. Once in the macro subsystem you may use any of the commands which are valid there to modify the current macro file or to create a new macro file.

Macro files are EWE files containing macro commands, which are new commands that you create by combining one or more existing commands. Only one macro file may be loaded into the system at a time. Loading a new macro file replaces the current macro definitions (called the current macro file) with the definitions contained in the new macro file.

More information about macros and the macro subsystem is presented in the macro subsystem section of this chapter.

PACK command

The PACK command allows you to reorganize the specified volume such that gaps left between existing files are minimized. This allows you to utilize all of the available space in any given volume. This command is active in all parts of the graphics editor.

PREFIX command

The PREFIX command allows you to change the default volume used by the graphics editor. This is only valid while in the graphics editor. Whenever a volume is not explicitly or implicitly stated, the system uses the default volume. The default prefix is set to EWSYS each time the Graphics editor is selected.

The prefix volume is useful in that you may set it to any volume. Thus, when storing data with the SAVE command, it is not necessary to specify the destination volume. When a file is being loaded into the system, the default volume is searched first. If the file is not found on the default volume, then the system looks on all other volumes for that file.

PROCESS command

The PROCESS command allows you to load a particular process file and/or enter the process subsystem. Once in the process subsystem you may use any of the commands which are valid there to modify the current process file or to create a new process file.

Process files are EWE files which define the units of a drawing (such as inches, feet, cemtimetres, etc.,) and the layers upon which the drawing is created and displayed. Layers are discussed, along with other concepts of the process subsystem in the process subsystem section of this chapter. Defining the layers in the process subsystem determines how the drawing is represented on the display. Only one process file may be loaded into the system at a time. Loading a new process file replaces the current process file with the definitions contained in the new process file.

REMOVE command

The REMOVE command allows you to remove unwanted files from any volume while you are in the graphics editor. There are three types of files which may be removed with this command: process files, macro files, and device files. Process files may be removed only when you are working in the process subsystem. Similarly, macro files or device files may be removed only when you are working in the macro subsystem or edit subsystem, respectively. Other types of files may be removed using the EWE Filer program.

RETRIEVE command

The RETRIEVE command allows you to create a device (a drawing) from the contents of an archive file. This is the reciprocal function of the ARCHIVE command. Entering the command causes the system to read the contents of the archive file and line by line reconstruct the device from which the archive file was created. If the device contains links to other devices, these linked devices are also reconstructed.

Entering the RETRIEVE command causes the system to enter the edit subsystem to create the device and any devices linked to it. As the device is created, the individual commands stored in the archive file are entered with the results displayed on the drawing area of the CRT (just as if you were entering the commands interactively from the edit subsystem). Once the device is reconstructed, the system saves the device(s) created and places the device files created on the specified volume.

You may stop the retrieve process at any time and either abort the command or make additional entries to the drawing by entering edit subsystem commands. The retrieve process may then be re-activated from the point it was interrupted by entering the INPUT command.

If the command is aborted, you are left in the edit subsystem with the partially reconstructed device displayed on the CRT. You may then use the commands of the edit subsystem to add to the device, modify the device, save the device, or exit the edit subsystem.

Entering the RETRIEVE command can cause the loss of some existing device files. When retrieving an archive file, the system first removes any device file on the destination volume whose file name matches the name of the device. Additionally, if the reconstructed device is linked to other devices, the system removes all device files on the destination volume whose file name matches the name of a linked device. This allows the system to store the drawing and linked devices on the destination volume with the same device file names with which they were originally stored. This eliminates any worry about creating duplicate device files.

VOLUMES command

The VOLUMES command causes the graphics editor to display a list of all volumes currently recognized by the graphics editor. The list shows each volume number and volume name. This command is active in all parts of the graphics editor.

Edit Subsystem

Overview

The edit subsystem is the portion of the Graphics Editor where drawings are created, modified, plotted, and stored. The drawing is displayed on the computer's graphic CRT and is stored on your mass storage device in an EWE file, called a device file.

You interact with the edit subsystem by typing commands, parameters, and data on the computer's keyboard or by using the graphics input tablet to select commands, parameters, and data from the graphic display. The latter method is the preferred method of entry as it allows you to see the effect of the command as it is being entered.

Some basic concepts are provided to help you understand how the subsystem works. The commands available in the edit subsystem are described, while several interactive examples are used to illustrate their functions.

Concepts

Devices

A device is a drawing. The drawing that you created in Chapter 1, EXAMP, is a device. Devices exist in computer memory and may optionally be stored in an EWE device file on your mass storage device. When a device is edited or modified, the changes are made to the copy of the device in computer memory. By storing the modified device you may change the copy of that device stored in the device file.

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

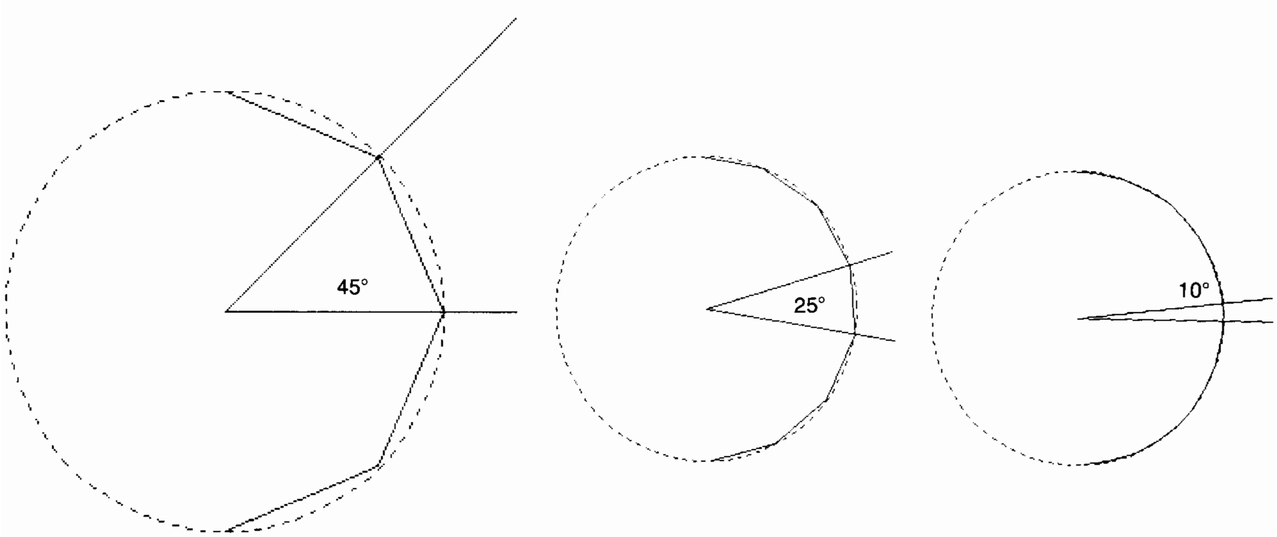
Components

Drawings are created in the edit subsystem by adding one or more components to a portion of the graphic CRT. A component is a basic geometric figure (such as an arc or a line), textual information, or a reference (link) to another device, called an instance component. Eight components are recognized by the system. These eight components are described next.

Arc Component

The arc component represents the geometric figure known as an arc. An arc component is represented by a section of a circle and is approximated by a series of equal length line segments. It is defined by two xy locations, called the endpoints of the arc, and a third point, which lies on the arc. This specifies a unique arc since three xy points define a unique circle, and the arc is simply a section of that unique circle.

The arc is approximated by a series of equal length line segments. By specifying the angle (called the resolution) in degrees between endpoints of these line segments, you can determine how smooth the curve of the arc appears. The drawings below show the relationship between the resolution of the arc and the smoothness of the curve of the arc.

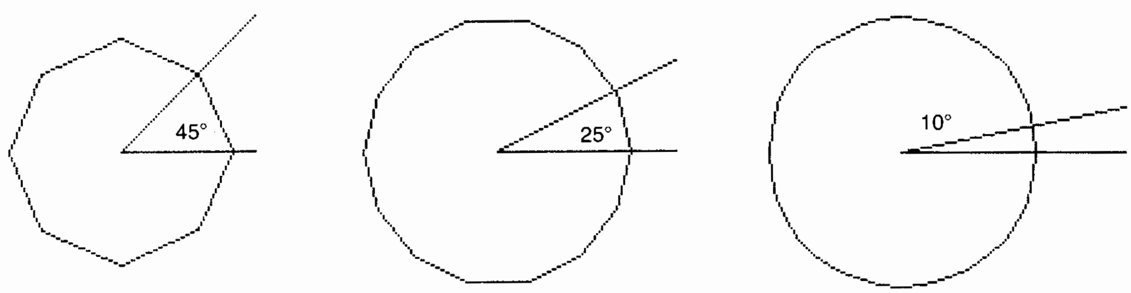


Resolution = 45   Resolution = 25   Resolution = 10

Circle Component

A circle component represents the geometric figure known as a circle. The circle component is approximated by a series of equal length line segments. It is defined by two xy locations, the center of the circle and any point on the circle.

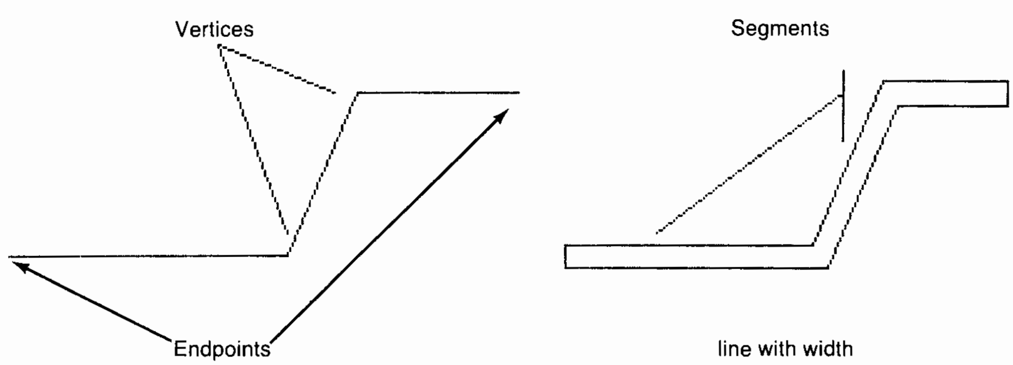
By specifying the angle (called the resolution) in degrees between endponts of the line segments approximating the circle, you can determine the smoothness of the curve of the circle. The drawings below show the relationship between the smoothness of the curve of the circle and the resolution of the circle.



Resolution = 45 | Resolution = 25 | Resolution = 10

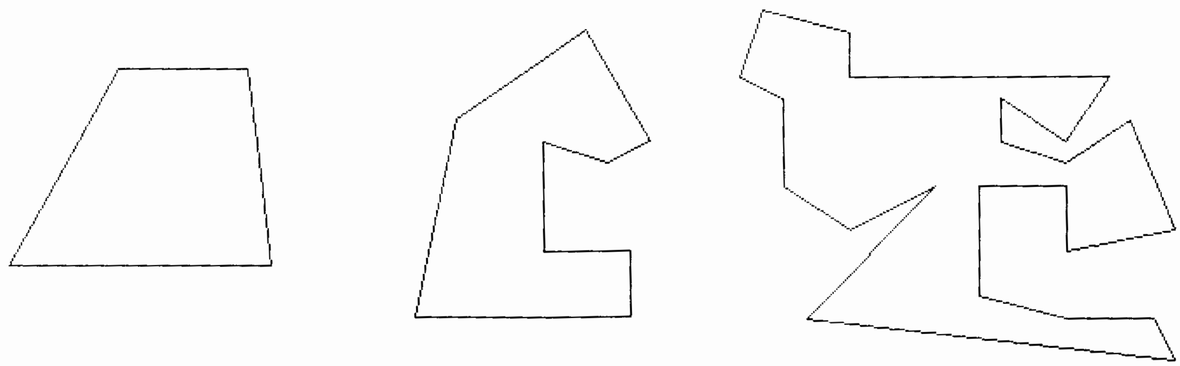
Line Component

A line component represents the basic geometric figure known as a line. The line component may be segmented and may have width. It is defined by two xy locations representing endpoints and up to 253 xy locations representing the junction of two segments, called vertices.



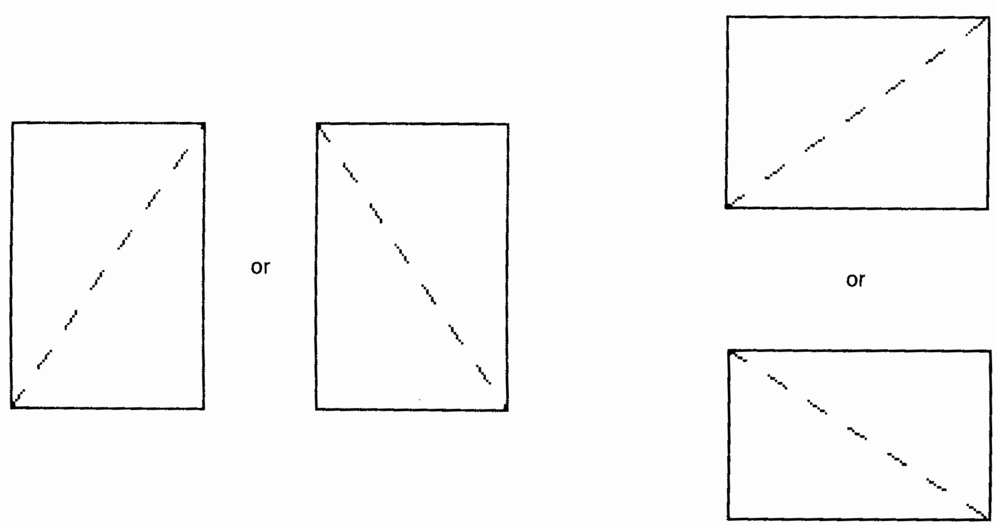
Polygon Component

The polygon component represents the basic geometric figure known as a polygon. It is a closed plane figure bounded by straight sides. A polygon component is defined by specifying the xy locations of its vertices in order. It may have a minimum of three and a maximum of 255 sides.



Rectangle Component

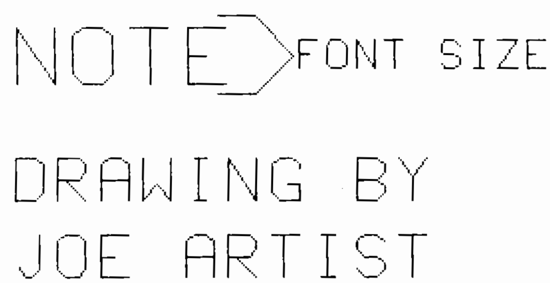
A rectangle component represents the basic geometric figure known as a rectangle. The rectangle is a parallelogram all of whose angles are right angles. Because of this special property, a rectangle component is defined by only two xy locations. These locations represent the locations of the endpoints of either diagonal of the rectangle, as shown in the drawing below.



Note Component

The note component is a stick-character representation of characters. You may specify the characters which compose the note component and control the size with which they are displayed on the CRT (called the font size). A maximum of 64 characters may be specified as a note component.

Note components are most useful for adding textual information to a drawing, such as the drawing creation date, the name of the draftsman, the title of the drawing, etc.



Text Component

The text component is a block-character representation of characters. You may specify the characters which compose the text component and control the size with which they are displayed on the CRT (called the font size). A maximum of 64 characters may be specified as a text component.

Text components are most useful for adding textual information to a drawing, such as the drawing creation date, the drawing number, the title of the drawing, etc.

Instance Component

The instance component is a reference (link) to another device. Linking a device to your drawing allows the components forming the linked device to be displayed in your drawing. This allows you to display components of a device in your drawing without re-entering the commands that formed the linked device.

For example, an electrical engineer creating a schematic drawing needs to represent 50 transistors in his drawing. Suppose that the transistor is represented by nine line components and one circle component. He could add the individual components that represent the transistor at each of the 50 locations in his drawing, requiring 50x10 or 500 individual adds of components. However, if he created a device named TRANS which represents the transistor, he would need to add the components forming TRANS only once. Then by adding instances of (links to) TRANS at the 50 locations, the components representing the 50 transistors are displayed in his drawing. Using instances of the device requires that you enter many fewer commands than adding each individual component, thus less memory and disc space are required for the drawing.

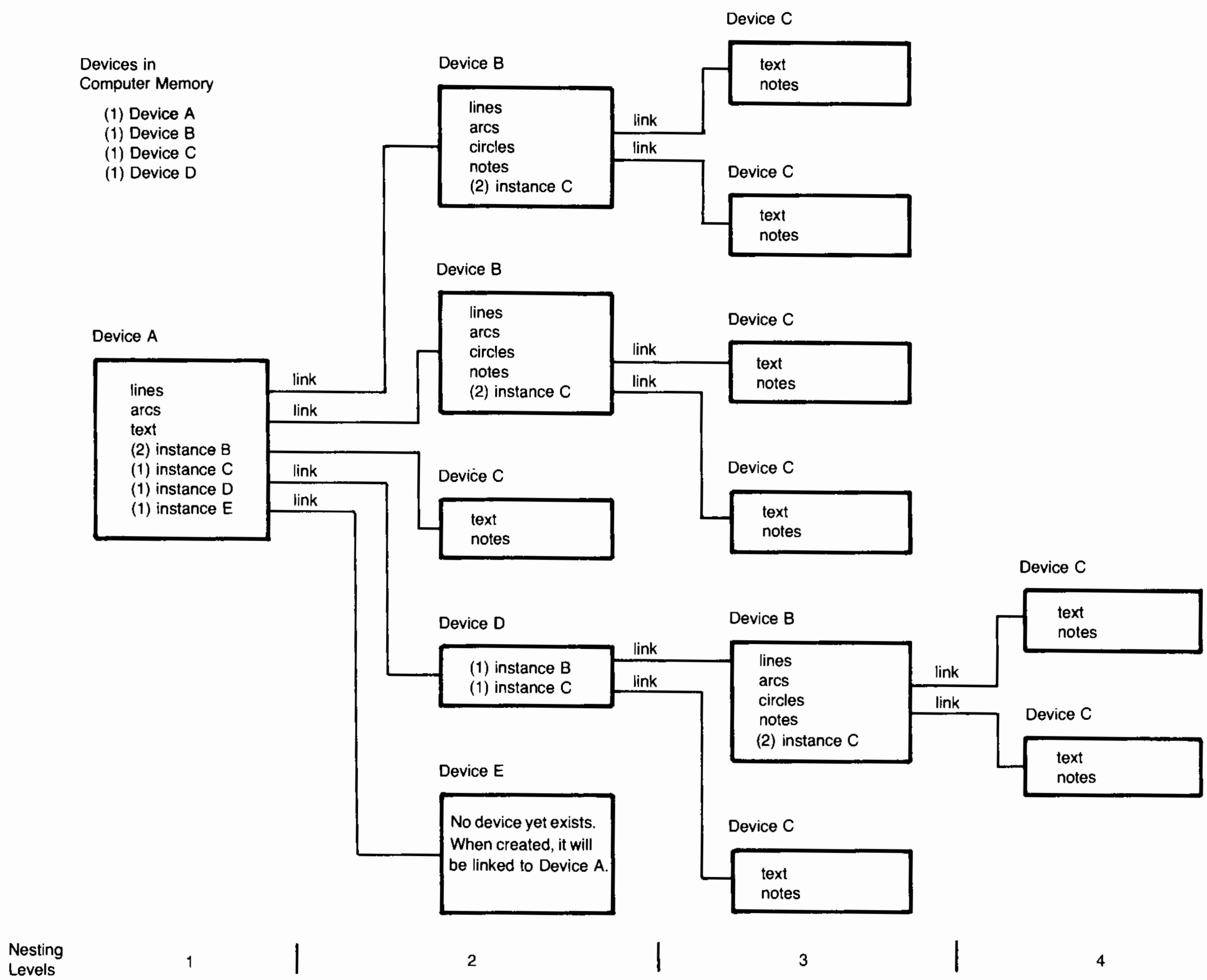
The components of the linked device remain separate from the components that form your drawing. Only the instance (link) is a component of your drawing, not the components that form the linked device. One copy of the linked device is resident in computer memory, regardless of the number of links to that device.

An instance may represent a link to a device that is as yet undefined. This allows you leave link(s) to the device in your drawing and create the linked device later.

For example, an architect creating a floor plan might not yet have decided which model and size of desk to use as office furniture. By adding instances of (links to) a device named DESK, he can create the floor plan without worrying about later adding the components of each desk. When he finally decides on the model and size of desk to use, he can create a device called DESK that accurately represents the particular desk he intends to use. This device is then displayed with his floor plan drawing at the locations specified by the instance components (links). No further modifications to the floor plan would be required.

If the architect later changes his mind as to the model and size of desk to be used, it is necessary only to modify the device DESK. Changing the device DESK would cause the floor plan to display the new device instead of the old, since the floor plan drawing contains only links to the device DESK, not the components that form it.

Just as your drawing may contain instance components that link other devices to it, the linked device may also contain instance components. Instance components and the links that they represent are shown in the drawing below.



Notice that the links divide the components displayed into segments. For example, device A contains four links to devices. These four devices are each one link away from the main device, device A. Similarly, some of these devices contain links to other devices. These devices are two links away from the main device, device A. The segments of the display are called nesting levels. Nesting levels allow you to control whether the components of a linked device are to be displayed in your drawing or not. This is discussed in depth with the nesting level concepts in this chapter.

Descriptors

A descriptor is simply a character or group of characters which describes to the system the type of component in which you are interested. Nine descriptors are recognized by the system, one for each of the eight components and an all-inclusive descriptor, specifying every type of component. You may define macros in the macro subsystem to replace one or more of these descriptors. Eight such macros are supplied with the system in the macro file, MACRDATA. When a descriptor is required with a command you may enter the appropriate descriptor from the table below or a macro that replaces the descriptor, providing that macro is currently defined in the macro subsystem.

Component	Descriptor	Macro (supplied in MACRDATA)
arc	A	ARCS
circle	C	CIRCLE
instance	I	(no macro supplied)
line	L	LINE
note	N	NOTE
polygon	P	POLY
rectangle	R	RECT
text	T	TEXT
every component	E	EVERY

Inames and Tnames

It is often very handy to attach a string of characters to a particular component that stays with the component wherever it is moved. For example, suppose that you add a rectangle component to your drawing that represents a desk. This may represent a particular model of desk, such as model 137. If the string of characters "model 137" could be attached to the rectangle component, there would never be any question as to which desk is represented in the drawing.

EGS-45 allows two such strings to be attached to each component. These strings are called the iname and the tname. The terms iname and tname arise from the fact that the IDENTIFY and TRACE commands respectively are used to display, change, and move the name.

You will find more information about inames and tnames with the TRACE, IDENTIFY, and ADD commands in the edit subsystem.

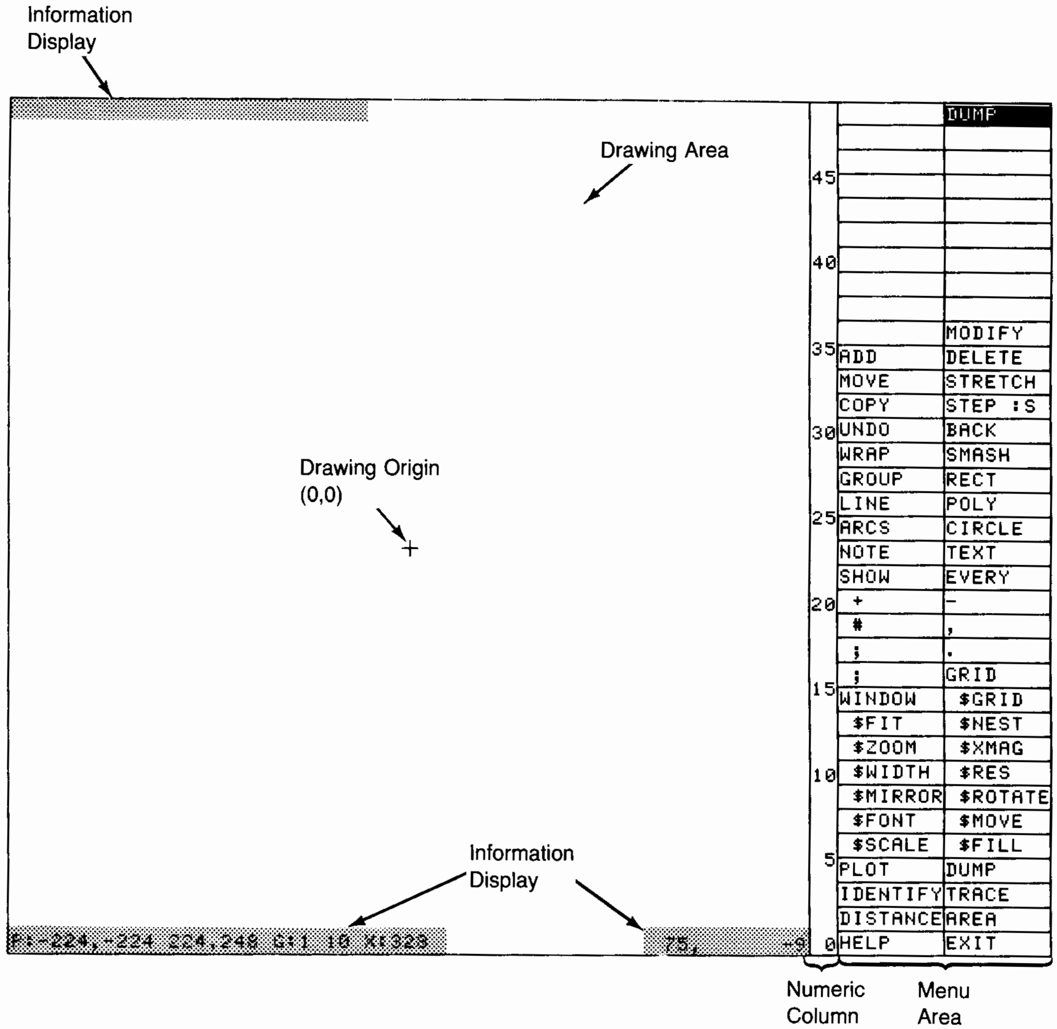
Library Parts

The term library part is simply another term for a device stored in a device file. You may add instances of a library part to your drawing. Adding an instance of a library part causes a copy of that device to be placed in computer memory. The instance links this device to your drawing allowing the components forming the device to be displayed with your drawing.

Modifying the copy of the device in computer memory does not change the library part, since it is stored in a device file. However, if the modified device is stored back in its original device file, the definition of the library part IS modified. Once a device is modified, all drawings containing links to that device, display the modified device.

The Display

When the edit subsystem is entered from the Graphics Editor, the graphic CRT should resemble the display below.



The display consists of four functional areas: the menu area, the numeric column, the information display, and the drawing area.

Menu Area

The menu display area consists of two columns of 35 menu slots, resembling small boxes. Each menu slot is capable of holding eight characters, which may be a command name, a parameter, a macro, or any other set of characters that is entered often. The information contained in the menu area is stored in an EWE file, called a menu file. The MENU command allows you to modify the contents of a menu slot or to load a different menu file.

The top right menu slot displays the command last entered in the subsystem and is reserved by the system for this purpose.

The menu area provides an alternative to entering edit subsystem commands and parameters from the computer keyboard. When a menu slot is selected, the contents of the menu slot is entered just as if it had been typed on the computer keyboard. Thus, the menu is simply a convenient alternative to typing.

Numeric Column

The numeric column area of the display allows you to enter numbers without having to type them on the computer's keyboard. Numbers are entered by selecting individual values from the column. For example, to enter the number 87, position the cursor over the number 8 in the numeric column and select; repeat this procedure for the number 7. You used this technique in the interactive example in Chapter 1 to enter angles of rotation and layer numbers. Similarly, to enter the number 146, you could: select the individual characters (1, 4, and 6) from the column, select the number 14 followed by the number 6, or select the number 1 followed by the number 46.

Information Display

The information display area of the CRT is indicated on the display representation above. The area in the upper left portion of the screen is used to display error messages and general messages issued by the system. The information displayed at the bottom of the screen provides valuable information about the current grid settings, and the current window settings. This information is discussed in detail with the window and grid concepts of the edit subsystem.

The Drawing Area

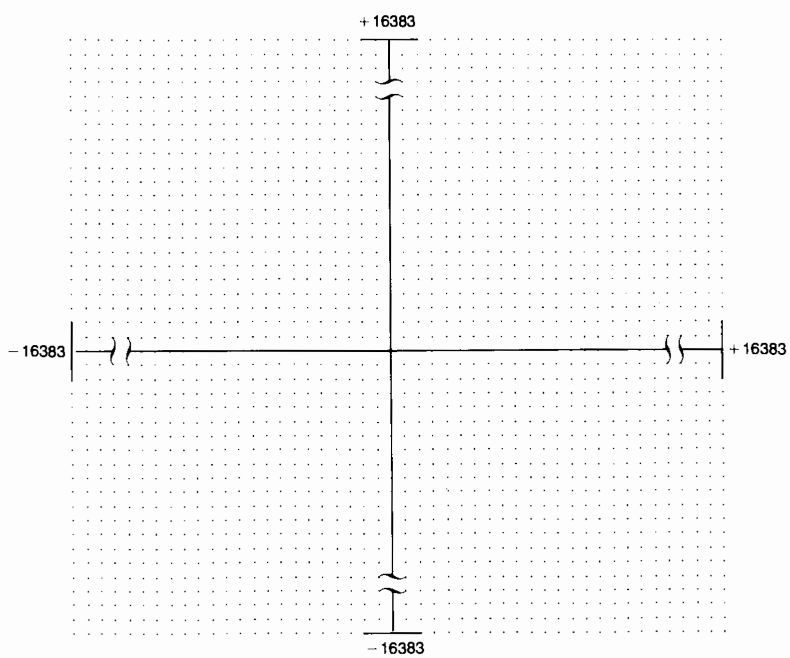
The drawing area of the CRT is a "window" looking over part or all of the total drawing area. This is the work space in which you create and modify drawings. With the WINDOW command you can change the location of this window (called panning), and thus change the portion of the drawing that is viewed through the window. Additionally, you may change the magnification of the portion of the drawing viewed through the window (called zooming). These concepts are discussed with the window concept later in this section.

To fully understand the drawing area you must first understand the concepts of layers and the system grid.

The System Grid

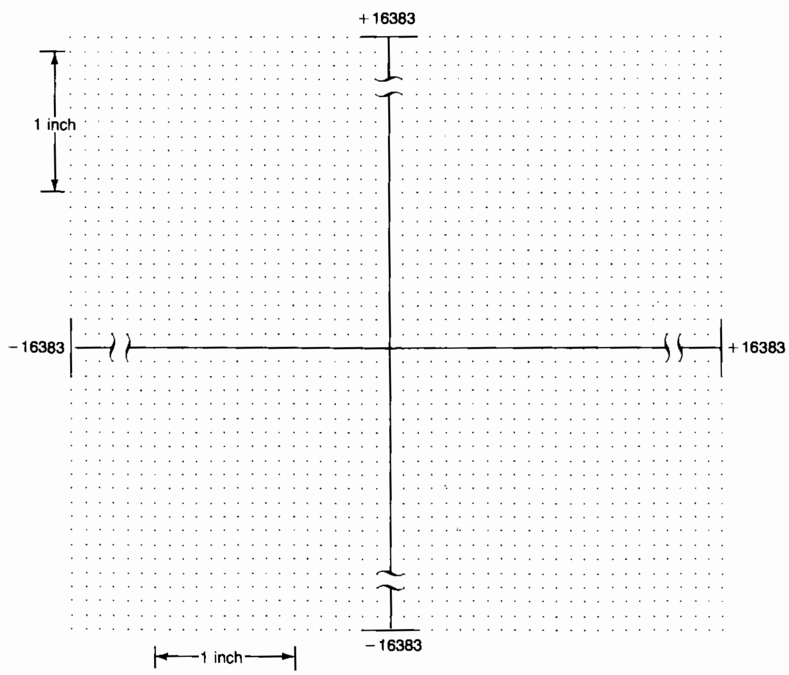
You are used to the idea of creating drawings on a sheet of paper with pen or pencil. With a sheet of paper you have a finite drawing area, bounded by the edges of the paper, in which you may place your pen at any location to draw. This gives you an infinite number of locations where you may place your pen.

With EGS-45 you also have a defined drawing area which is referred to as the total drawing area. 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 an integer value (see the drawing below). The array of these locations is called the system grid. There are a total of 32 767 system grid points (− 16383 to + 16383) in both the X and Y directions.



User Units

By using the UNITS command in the process subsystem you may assign units (micrometres, centimetres, mils [.001 inch], inches, or feet) to the system grid, thus allowing you to work in units that make sense in your application. For example, you can specify that the units of the drawing is 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 that you can specify in your drawing is .1 inch. This is illustrated in the drawing below.



The User Grid

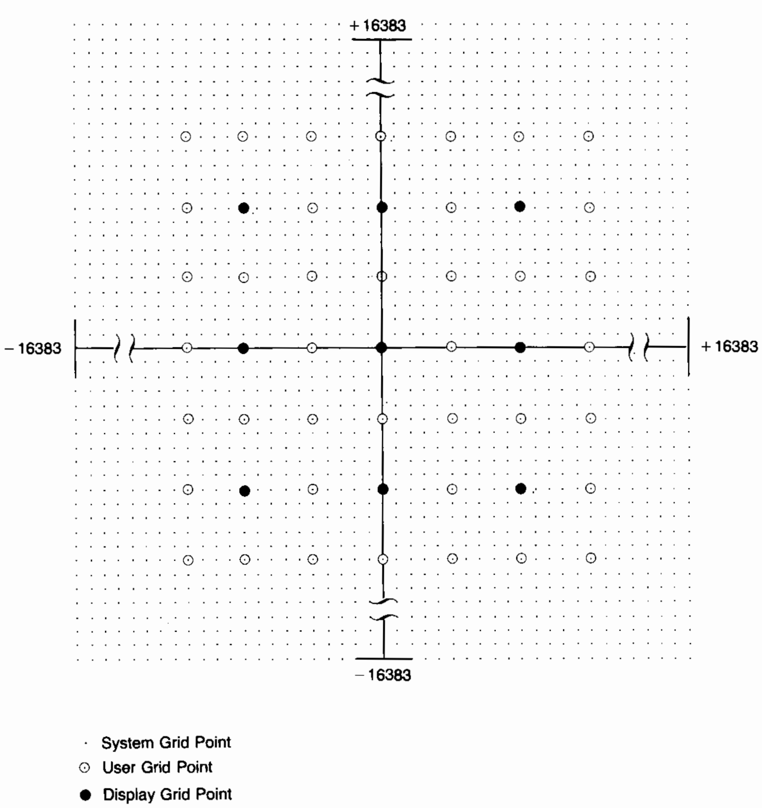
With the GRID command, you can limit the number of system grid points that are active in the drawing area of the display, thus declaring a subset of the system grid to be active. This subset of the system grid is called the user grid.

In the example with the user units concept, the units of the system grid are set such that there is .1 inch between adjacent system grid points. When entering locations, such as the location of an endpoint of a line that you are adding to the drawing, the location entered is snapped (shifted) to the nearest user grid point. When these points are close together, it is difficult to select the exact grid point you want. For example, when specifying the location 0,1.0, it is difficult to keep from specifying the point 0,0.9 or 0,1.1.

With the GRID command, you can define a subset of the system grid, called the user grid, by disabling a number of the system grid points. For example, you could define a subset of the .1 inch system grid such that a user grid point exists every .5 inch. Thus, although a system grid point is defined every .1 inch, displaying only user grid points makes it much easier to add components to the drawing where the resolution of the system grid (.1 inch) is not needed.

The Display Grid

The GRID command allows you to limit the number of user grid points that are displayed in the drawing area, thus keeping the drawing area uncluttered. This 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 drawing below.

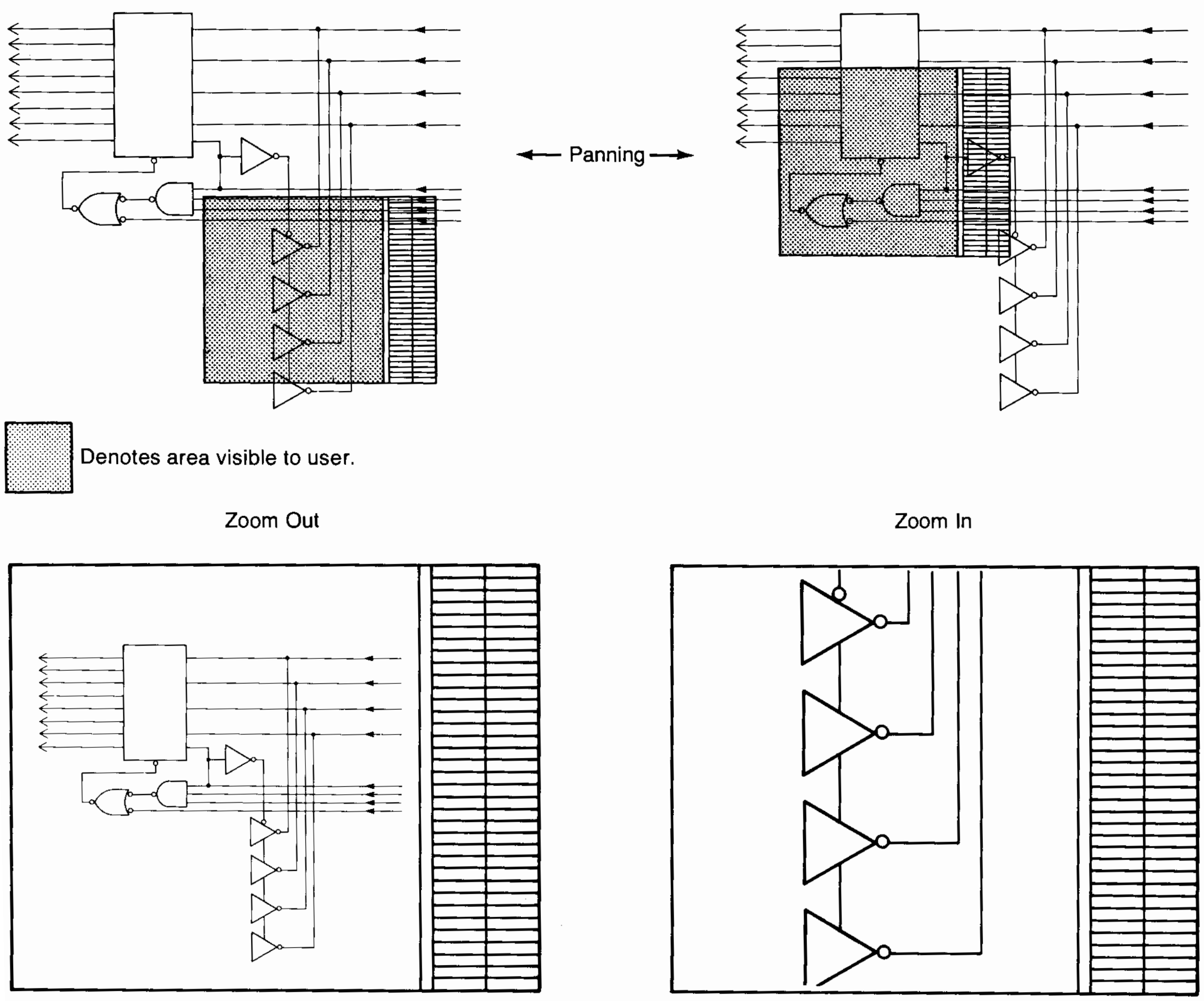


The two values adjacent to the characters G: in the information display portion of the CRT provide information about the current user grid and display grid settings.

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. Thus, a user grid point exists every .5 inches. The second value specifies the number of user grid point to be displayed. Thus, every 2nd user grid point is displayed. Alternately stated, each grid point displayed is 1 inch apart (two user grid points times .5" per user grid point).

The Window

The drawing area of the CRT is a "window" looking over the total drawing area or any portion of it. Several commands are provided in the edit subsystem which allow you to control the location of this window and the magnification of the drawing viewed through it. Shifting the position of the window left, right, up, and down is called panning. Moving the location of the window nearer or farther from the total drawing area (and thus changing the magnification of the drawing viewed through the window) is called zooming.

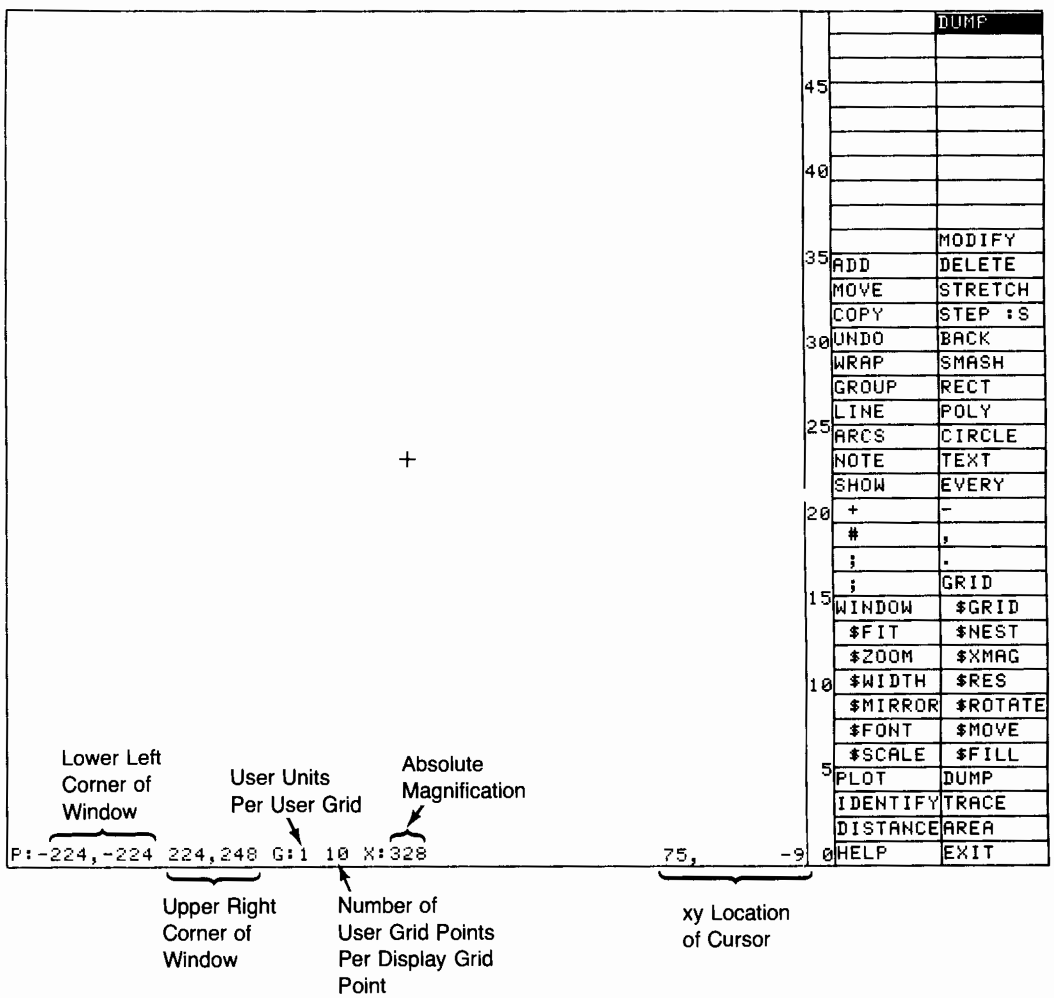


The information contained in the lower left corner of the CRT, adjacent to the characters P:, displays the location of lower left corner and upper right corner of the window, in user units.

The ratio of height to width of this window area is 453/430. This means that if the window size is .906 inches in the Y direction, its size is .860 inches in the X direction. Whenever the window location or magnification is changed, this ratio is maintained.

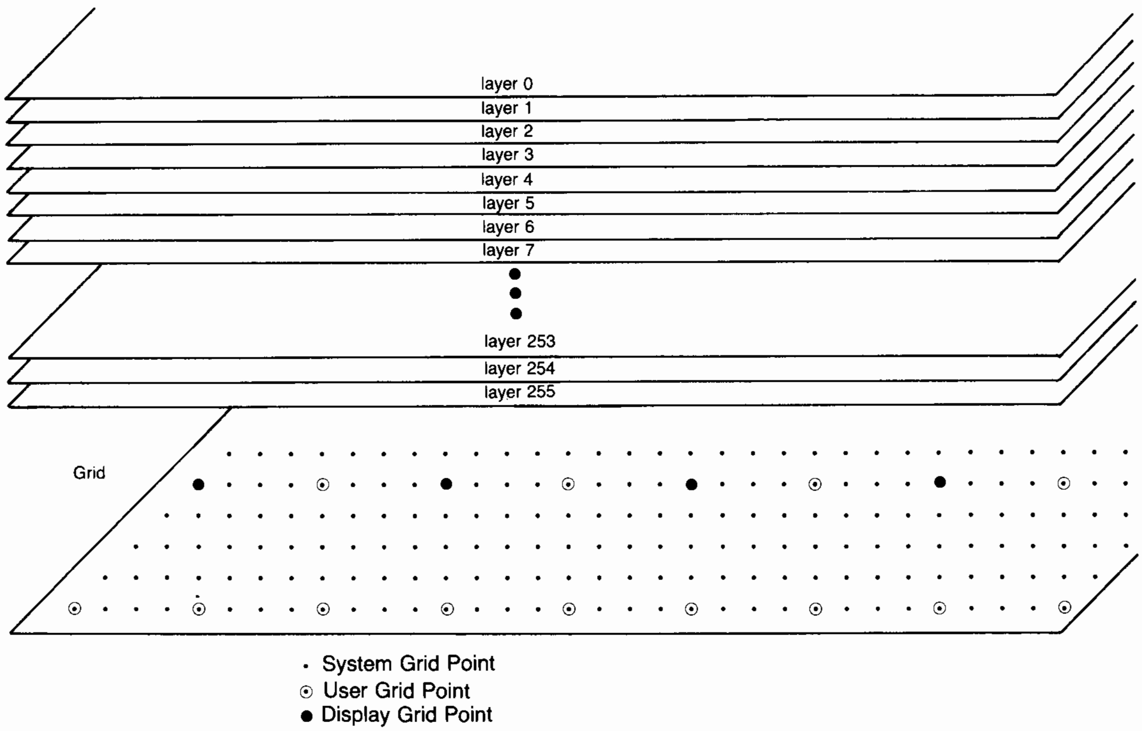
The value adjacent to the characters X: in the lower center of the window indicates the absolute magnification of the drawing viewed through the window. For example, X:2 specifies that 1 inch of your drawing (as defined by your user units) is displayed as 2 physical inches of the display.

The values displayed in the lower right corner of the window indicate the location of the cursor, in user units. As you move the location of the cursor with your graphic input device, you should observe these values change and reflect the new cursor location.



Layers

So far, you have seen that the entire drawing area is defined by the system grid, the UNITS command in the process subsystem, and the GRID command in the edit subsystem. The total drawing area can be thought of as a stack of 256 clear plastic sheets, called layers, upon which the drawing is created.



When creating a drawing, you add components to one or more of these layers (in the interactive example in Chapter 1, you entered commands such as ADD LINE 1 ... which specifies adding a line to layer 1). Layers can be viewed individually, or in combination with other layers. In the interactive example, you were viewing all 256 layers at once. By using the SHOW command in the edit subsystem, you can specify which layers are to be displayed. In this fashion, you can enter an entire drawing and view only those portions that you want to see.

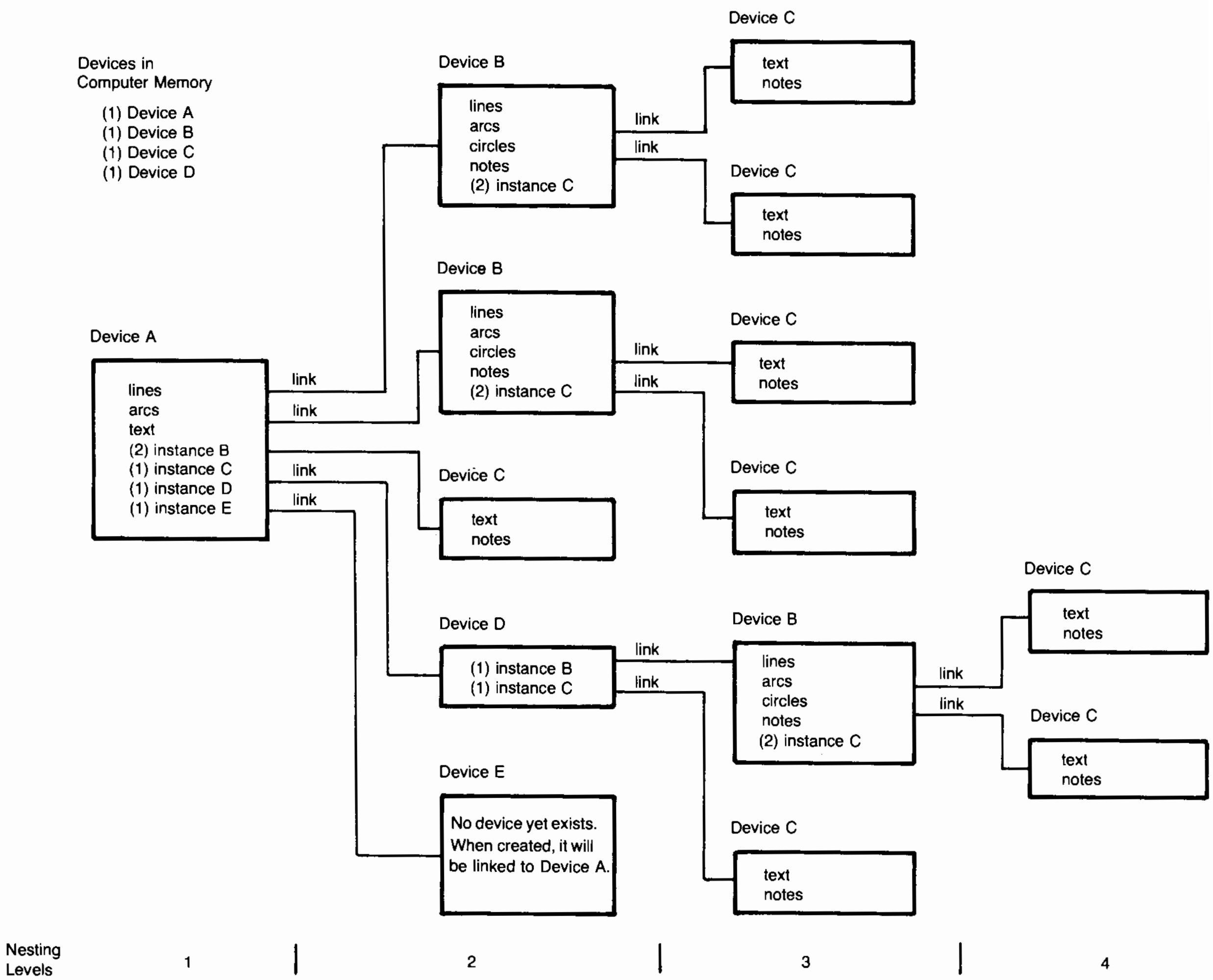
For example, an architect may create a floor plan that includes lighting fixtures, plumbing outlets, and furniture placement. By entering the furniture placement on layer 1, the plumbing outlets on layer 2, and the lighting on layer 3, the architect can view only the lighting, only the plumbing, only the furniture placement, or any of these in combination with the other(s). Thus from only one drawing on the system, any number of useful drawings can be displayed and plotted.

Before a layer may be used, it must be defined in the process subsystem. Defining a layer involves, among other things, specifying the line type (solid, dashed, etc.) with which components entered on that layer are to be displayed. Additionally, one of three layer types must be declared with the layer definition: detail layer type, symbolic layer type, or interconnect layer type. Different layers may be assigned different layer types. The layer types correspond to the display modes of the CRT: detail mode, symbolic mode, and interconnect mode. When a display mode is set, only components entered on the corresponding layer type(s) are displayed. This is discussed in more detail with the process subsystem concepts.

Layer 0 is reserved by the system and must always be defined. Layer 0 is used to display instance components, when the instance component is present at the deepest nesting level displayed. Also, layers 1-100 are reserved for factory defined process files.

Nesting Levels

When examining the concepts of instance components in this chapter, you found that the links to other devices divide your drawing into segments, called nesting levels. This is indicated in the drawing below.



The instance component allows the components of the linked device to be displayed with your drawing. Nesting levels allow you to specify which device's components are displayed with your drawing. With the WINDOW command you may specify a maximum nesting "depth", such that only the components of devices in nesting level 1 to the specified depth are displayed.

For example, specifying a maximum nesting depth of 2 with the WINDOW command for the drawing above would cause the drawing to show the components of device A. The instance components in nesting level 1 are replaced with the components of the devices to which they link. The instance components of devices B and device D in nesting level 2 (the maximum displayed nesting depth) are represented as rectangles in layer 0. Instance components at the maximum displayed nesting depth are ALWAYS displayed as rectangles in layer 0.

The commands of the edit subsystem (such as STRETCH, COPY, and DELETE) can act only on components in nesting level 1. To modify components of devices in deeper nesting levels, it is necessary to use the SMASH command to bring the components of the device into nesting level 1.

EGS-45 can deal with a maximum of 500 nesting levels.

Logical Levels

Through the use of logical levels, a user is able to specify a hierarchy of instances within a drawing. This feature is useful when using the GENERATE command to eliminate any unnecessary instances in the generate process.

The logical hierarchy is simply an assignment of a logical level number to each instance. Higher positive numbers mean a higher position in the hierarchy. Instances with a lower logical level number can be used to create another instance with the same or higher level number, but cannot be used to create lower level instances. The default level number of one is assigned to all instances at time of creation. The level number can only be changed by editing the instance as an individual drawing. If an instance created by the use of the WRAP or GROUP command requires a different logical level number than the default of one, it has to be stored and then edited as an individual drawing before the LEVEL command can be used to change the logical level number.

Instances that are specified as part of a drawing are checked for a logical level that is equal to or less than the current drawing. An error message is generated if this condition is not met. An example of an error would be to use an instance that was defined with a logical level number greater than the default logical level of a new drawing. Another example would be to change the logical level of an instance to a higher number and then edit a drawing that uses the modified instance. If the modified instance logical level exceeded the current drawing logical level, an error would be displayed at time of loading.

Command Entry

Commands are key words (and parameters) which tell the computer what you want to do. Commands are entered in the edit subsystem either by the computer's keyboard or by a graphic input device. When commands are entered with the graphic input device (this is the prefered method), you use the stylus of the graphic input device to position the CRT cursor over the menu slot containing the command that you want to enter, and select. Parameters and data may be entered with the command by selecting the parameter from a menu slot and the data (such as the layer number or the angle of rotation) from the numeric column.

A command must be terminated before it is executed by the system. Typically, a command is terminated with the semicolon character (;), however beginning a new command before terminating the previous command also terminates the previous command. A few commands require no specific terminator. This is shown in the syntax diagrams in the syntax reference section.

Commands in the edit subsystem may also be entered by typing the command name and any parameters and data on the computer keyboard. The information you enter is echoed on the computer CRT as it is typed. Only the first three characters of a command name need be entered (for example, entering WIN is equivalent to entering the entire command name, WINDOW). Additional characters of the command name may be entered; however, they must be the correct characters in the correct order. For example, WIN, WIND, and WINDO are all correct entry for the WINDOW command. However, WINDW is not a correct entry for the WINDOW command because the character W should have the character O before it.

Spaces or commas are recognized as delimiters by the system and should be used to separate adjacent parameters (:L,:F or :L :F) and data (100,100 or 100 100). If these delimiters are omitted when required, an error or unexpected result may occur.

Pressing either the CONTINUE key or the STORE key tells the system that you are finished typing the command, thus allowing input to continue from the graphic input device. Pressing the EXECUTE key has the same effect as pressing either STORE or CONTINUE, however a semicolon (;) is automatically appended to the end of the characters entered. Since many of the commands in the edit subsystem require a semicolon at the end of the command, this is a handy feature.

A command may be entered using both the graphic input device and the computer keyboard. When entering information from the computer keyboard, you may wish to switch to the graphic input device (for example, you might type ADD PART :R90 and then switch to the graphic input device to specify a location in the drawing area). To do this, when you are finished typing, simply press the STORE key or the CONTINUE key (pressing the EXECUTE key enters a semicolon, thus terminating the command). When entering information with the graphic input device and you wish to switch to the computer keyboard (for example, to type a device name or the characters forming a note component), simply start typing.

Current Command

As a command is entered, it becomes the current command. The current command name is displayed in the upper right menu slot. The system remembers the command and any parameters and data entered with it. If you wish to enter the 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 TEXT 1 $FONT 100 $ROTATE 90 'TEXT 1' x,y ;
```

This adds the text, TEXT 1, to the drawing with a font size of 100 user units and a rotation of 90 degrees at the location x,y. To add the second text component with the same angle of rotation but a font size of 200, it is only necessary to enter the following because of the concept of current command:

```
$FONT 200 'TEXT 2' x,y ;
```

To add a text component identical to the last text component (TEXT 2) at a new location, x2,y2 it is only necessary to enter the following:

```
x2,y2 ;
```

The command, descriptor, layer number, font size, angle of rotation, and characters forming the text component are all remembered as the current command.

If you wish to enter a new descriptor or layer number for the current command, both the descriptor and layer number must be entered.

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

Suggested Procedures

Before entering the edit subsystem to create a drawing or modify an existing drawing, there are several procedures that, if followed, can ease your task. These procedures are listed and explained below.

- Select the File Utilities module and obtain a printed listing of the files present on all volumes currently on line. This provides a list of all device files on the volumes and tells you which existing library parts you can use to modify/create the drawing.
- Use the PROCESS command in the Graphics Editor to load the process file you wish to use (unless you intend to use the auto-load process file, PROCDATA).
- Use the MACRO command in the Graphics Editor to load the macro file you wish to use (unless you intend to use the auto-load macro file, MACRDATA).
- Use the LIST command in the Graphics Editor to obtain a printed listing of the contents of the macro and process files.

Having a printed listing of the layer definitions allows you to quickly determine which layer to use when creating your drawing.

Once in the edit subsystem, the following procedures should be among the first performed:

- Use the MENU command to load the menu file you wish to use (unless you intend to use the auto-load menu file, MENUDATA).
- Use the GRID command to determine your user grid and your display grid.
- Use the WINDOW command to move the window to the location you desire and set the magnification of the drawing viewed through the window.

Edit Subsystem Commands

The commands of the edit subsystem allow you to create, modify, store, and plot drawings. For conceptual purposes, these commands can be divided into four functional groups: drawing commands, screen control commands, modification commands, and information commands. However, the system makes no distinction between the different command groups. The system software diagram at the beginning of Chapter 1, shows the commands available in the edit subsystem.

Commands in the edit subsystem often require x,y locations to be entered. The locations are entered in user units and snapped (shifted) to the user grid point nearest the location entered.

The general function of each of the commands of the edit subsystem is discussed in the sections below. Additionally, after each group of edit subsystem commands are discussed, an interactive example is provided, similar to the example in Chapter 1. This is done to give you a first exposure to the command and to help present the concepts presented earlier in this chapter.

The syntax for each of the commands is found in the syntax reference section. The syntax reference section details the methods used to exercise the capabilities offered by each command.

Drawing Commands

Drawing commands are used to create a drawing by adding components to the drawing area of the CRT. They allow you to add components and instances of library parts to the drawing area, create library parts from components in the drawing, and store the drawing for later use.

ADD command

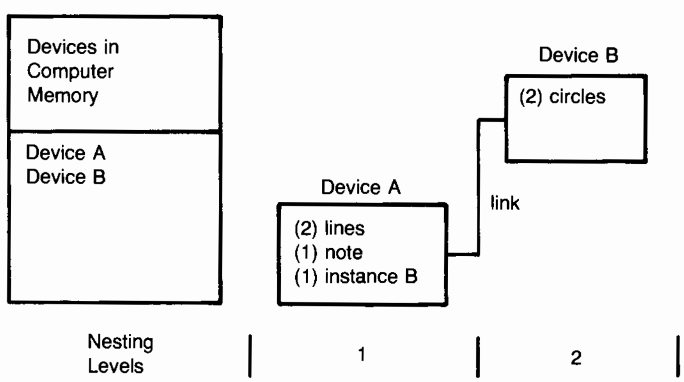
The ADD command allows you to add any component to the drawing area of the display. Many different options may be specified with the ADD command, depending on the component being added. A descriptor and layer number must be specified when adding a component (except for instance components, which are identified by the name of the device to which they link, and are represented with a rectangle in layer 0 when the instance is in the deepest nesting level displayed).

WRAP command

The WRAP command allows you to create a device from components in the drawing area of the display. All components located totally within a rectangle defined by two x,y locations entered with the WRAP command, become the new device.

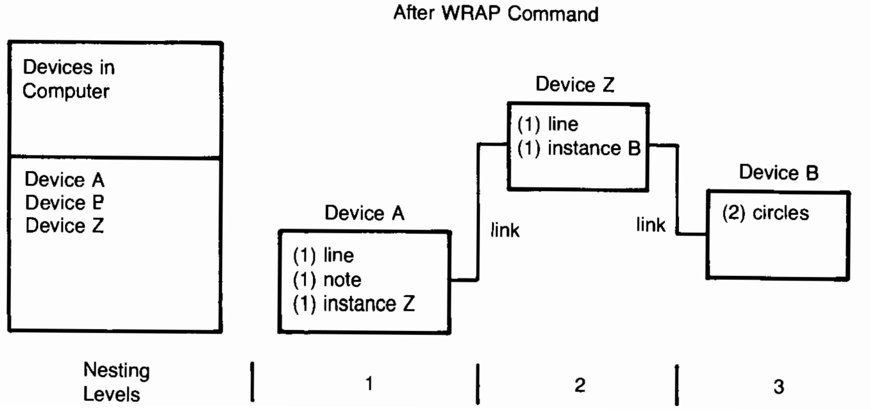
The components forming the device are removed from the drawing and are replaced by an instance component in nesting level 1 (thus linking your drawing to the new device). The components forming the device are displayed in nesting level 2 on the same layers that they were placed in the drawing. A copy of the new device is placed in computer memory.

Before WRAP Command



WRAP Command used to create Device Z, from 1 line component and 1 instance component of Device B.

After WRAP Command



Devices created with the WRAP command must be assigned a name if you wish to be able to add instances of the device to a drawing and if you wish to save the device as a library part for later use. Named devices created with the WRAP command are stored on the mass storage device when the drawing in which they are created is stored. If the drawing is not stored, the named devices created during that edit session can still be stored (thus becoming library parts) by exiting the edit subsystem, editing the device, and finally saving the device.

When the new device is created, you must assign a reference point which allows accurate placement of instances of the device (such as with the MOVE, COPY, or ADD commands).

GROUP command

The GROUP command allows you to create a device from components in the drawing area of the display. Individual components are selected to be included with the new device by specifying an x,y point located in or on the component. A descriptor and layer number may be specified to limit component selection to a specific component on a specific layer. Entering a device name limits component selection to instances of that device.

The components forming the device are removed from the drawing and are replaced by an instance component in nesting level 1 (thus linking your drawing to the new device). The components forming the device are displayed in nesting level 2 on the same layers that they were placed in the drawing. A copy of the new device is placed in computer memory.

Devices created with the GROUP command must be assigned a name if you wish to be able to add instances of the device to a drawing and if you wish to save the device as a library part for later use. Named devices created with the GROUP command are stored on the mass storage device when the drawing in which they are created is stored. If the drawing is not stored, the named devices created during that edit session can still be stored (thus becoming library parts) by exiting the edit subsystem, editing the device, and finally saving the device.

When the new device is created, you must assign a reference point which allows accurate placement of instances of the device (such as with the MOVE, COPY, or ADD commands).

COPY command

The COPY command allows you to make a copy of any component located in the drawing area of the display and place that copy at any location in the drawing. Copies are made by first identifying the component you wish to make a copy of. This is done by specifying a point that lies in or on the component to be copied. An optional descriptor and layer number may be used to limit selection of the component to be copied.

Next, you specify a reference location so that you may accurately place the copy of the component. Finally, you specify x,y location(s) where you wish to place copies of the component. The copies are placed such that the reference point of the component is placed exactly on the placement point.

STEP :S command

The STEP :S command, also called the step and repeat command, allows you to make multiple copies of component located in the drawing area of the display. The multiple copies that are created are placed in an array-like fashion in the drawing area. You determine the number of copies by specifying the number of rows and columns in the array of copies. For example, specifying a 3 column by 4 row array of copies causes 12 copies to be placed on the drawing, where the copies are arrayed with 3 copies per row, and 4 copies per column. You specify the x and y spacing between the copies in the array.

Multiple copies are made by first selecting the component that is to be copied. It is specified by entering an x,y point that lies in or on the component. An optional descriptor and layer number limit the selection of the component to be copied. Entering a device name limits component selection to instances of that device.

Next a reference point is specified on the component. This reference point allows the copies of the component to be accurately placed, as the copies are added at a location such that the reference point is added exactly on the location specified. An x,y point is entered which specifies the x and y distances between reference points of the copies.

INPUT command

The INPUT command allows you to read the contents of a generate file and add to your drawing the components of the drawing stored in the generate file. Entering the INPUT command with the name of a generate file causes the system to read the ADD commands contained in the generate file. Next the system enters those commands in the edit subsystem, just as if you were typing them on the computer keyboard or selecting them from the menu area of the display.

The INPUT command may be aborted, or temporarily suspended at any time during the input process. If the command is aborted, no further input occurs. If the command is suspended, you may make additions to the drawing being created from the generate file by entering edit subsystem commands. Once these commands are entered, the INPUT command may be re-activated such that input continues from the point at which it was suspended.

LEVEL command

The LEVEL command is used to assign the logical level number to a drawing. The logical level allows a user to specify a hierarchy within a drawing. This hierarchy may then be used in the GENERATE command to output drawing commands down to a specified logical level.

An example for using the levels would be in designing an electrical schematic. Base level library parts are designed and assigned LEVEL 1. Complex sets of the library parts are assigned LEVEL 2. Finally, a schematic drawing is assigned LEVEL 3. Thus, you have established a logical hierarchy to the drawing which is user-definable. When it comes time to generate this drawing using the generate command, you may select the level to which the generator will be done. If LEVEL 1 is selected for this particular drawing, everything in the drawing will be smashed except parts defined as LEVEL 1. They will be referenced, but not smashed.

DUMP command

The DUMP command allows you to copy the graphic CRT to the computer's built in printer.

PLOT command

The PLOT command allows you to plot a drawing on the current system plotter. Only portions of the drawing in active layers (those that have been turned on with the SHOW and WINDOW commands) are plotted. Plots may be scaled or not scaled.

If a plot is not scaled, the system plots the portion of the drawing displayed in the drawing area on the plotter. The lower left corner of the window is "mapped" to the lower left corner of the plotter (P1 on HP plotters). The drawing in the display area is plotted such that it occupies the largest area possible on the plotter (defined by P2 on HP plotters) without distorting the drawing. This means that the ratio of the height to width of the drawing area (453/430) is maintained.

If a plot is scaled, the drawing is plotted to the current scale (as specified by with the UNITS command in the process file) multiplied by the scale factor supplied. For example, suppose that the units of a drawing is feet and the drawing ranges from -20 feet to $+ 20$ feet, in user units. If the drawing is plotted with the scale factor 2, the system attempts to plot the drawing from -40 feet to $+ 40$ feet, an obvious limitation for a 11" x 17" plotter, such as the HP 9872. The system attempts to plot all of the drawing located in the positive X and Y directions from the current location of the lower left corner of the window, whether displayed or not. Portions of the drawing exceeding the plotter limits (P1 and P2) are not plotted.

The optional parameter causes both a scaled and an unscaled plot to be centered on the plotter. Another optional plot parameter allows you to change the resolution for all circles and arcs.

A fill option to the PLOT command, causes all circle components, rectangle components, and line components with width to be filled solid as they are plotted.

You may specify the pen speed with which the drawing is to be plotted and the width of the pen being used to create the plot. The width of the pen is important when filling components. If the actual pen width is less than the width specified, portions of the filled components will not be filled. If the actual pen width is more than the width specified, the components will be "over filled", thus changing their size.

SAVE command

The SAVE command allows you to store the drawing currently being modified in the edit subsystem, in an EWE file called a device file. When a drawing is stored, all named devices linked to the drawing which were created or modified during that edit session, are saved also. Each linked device is stored in its own device file. Links to unnamed devices are smashed, thus bringing the components forming the device into the next higher nesting level so that the components are saved with the named device referencing them.

EXIT command

The EXIT command allows you to exit the edit subsystem and returns you to the Graphics Editor.

If the drawing you are editing has not been saved before the EXIT command is entered, the system issues a message to that effect. You may then either save the drawing, modify the drawing further, or re-enter the EXIT command to exit the subsystem without saving the drawing. You may later re-enter the edit subsystem with the EDIT command to save any named devices that have not been saved.

A Simple Example

Before discussing the capabilities of the other edit subsystem commands, a simple interactive example is presented. By working along with this example, you will gain a better understanding of the concepts of the system. Additionally, by following the syntax diagram for each command (located in the Syntax Reference), you can gain an understanding of the different options offered by each command.

The drawing that you create, represents the floor plan of a small office complex. You will use the office floor plan drawing that you created in Chapter 1 as part of this drawing.

As you work the example, use the technique of selecting to enter commands, parameters, and data from the menu area of the display. If you make an error entering a command, refer to the error recovery techniques described with the interactive example in Chapter 1.

To create the example drawing, perform the following steps.

1. Load and run the EGS-45 software.
2. Select the Graphics Editor from the menu on the CRT.
3. List the contents of the current process file by entering:

   ```
   LIST PRO ;
   ```

   This list tells you the current user units. Additionally, it tells you the definitions of all currently defined layers.
4. Enter the edit subsystem by entering:

   ```
   EDIT ;
   ```

5. Define the user grid to be 40 user units per user grid point and display every user grid point. To do this, enter:

   ```
   GRID 40,1 ;
   ```

   To display the grid, enter:

   ```
   GRID ;
   ```

6. Change the position of the window by entering the lower left and upper right corners. Enter:

   ```
   WINDOW -1120,-1168 1120,1192 ;
   ```

7. Add a rectangle to your drawing that represents the entire floor area of the office complex. Enter:

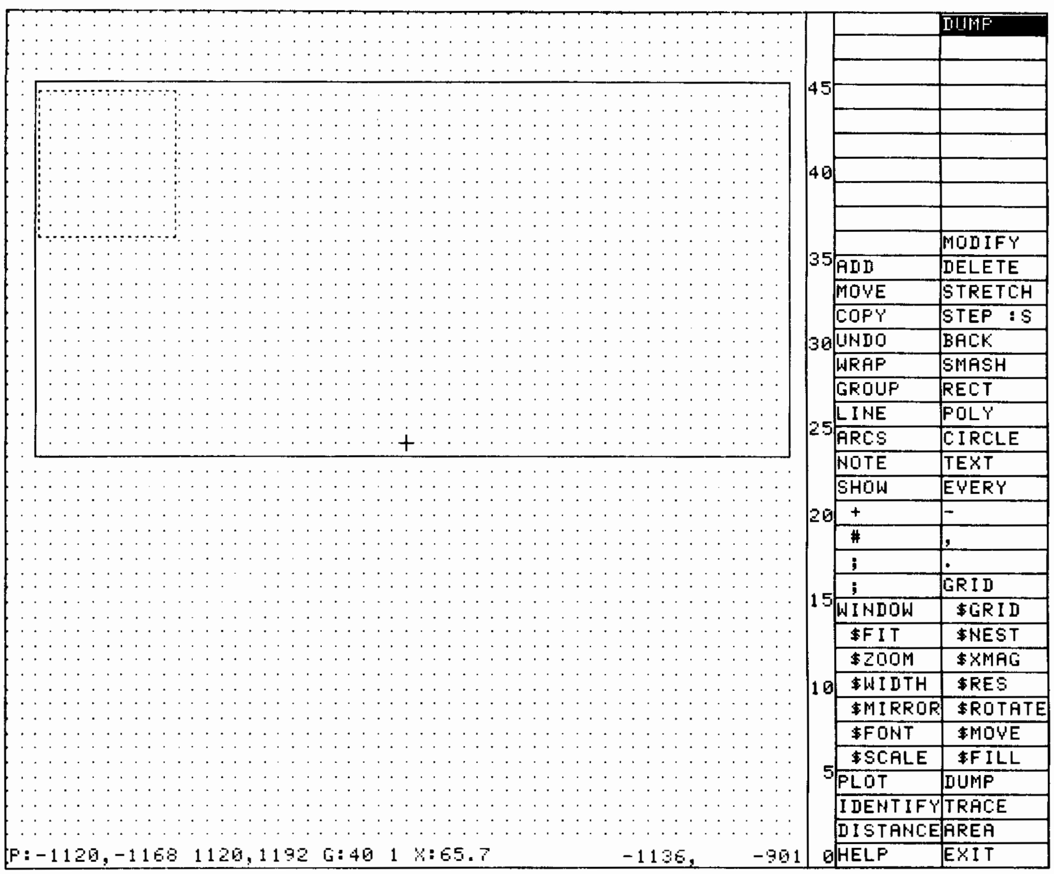
   ```
   ADD RECT 1 -1040,-40 1080,1000 ;
   ```

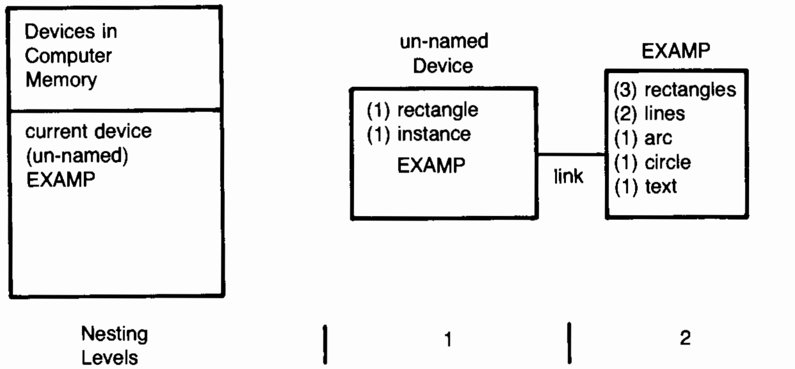
   The user grid is set such that 40 user units equals one user grid point. Since any point entered is snapped (shifted) to the nearest user grid point, it is not necessary to select the exact value. If the value you select is within 20 user units of the designated location, the selected point is snapped to the correct user grid point.

8. To represent the individual office plans in the floor plan layout, link the office plan drawing that you created in Chapter 1 to your current drawing. To do this, enter:

   ```
   ADD EXAMP -840,760 ;
   ```

   The instance component (link) is displayed in your drawing as a rectangle in nesting level 1, the deepest nesting level currently displayed. The components of the linked device are displayed in nesting level 2.





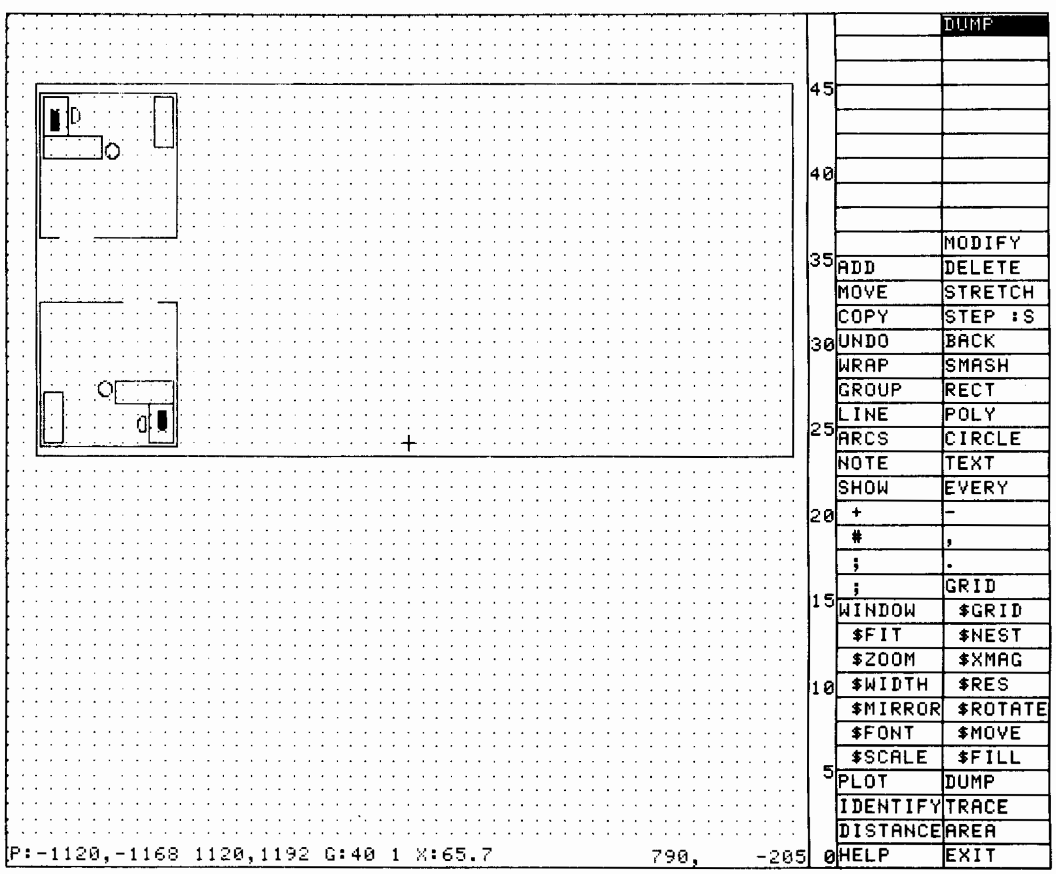
9. To view the components in nesting levels 1 through 2, enter:

```
WINDOW $NEST 2 ;
```

10. Place another copy of the office plan in the lower left corner of the floor plan. To make sure that the door to this office opens on an aisle, the instance will need to be rotated by 180 degrees. To do this, enter:

```
ADD EXAMP $ROTATE 180 -840,200 ;
```

The link to this device is contained in nesting level 1, while the components of the linked device are displayed in nesting level 2.

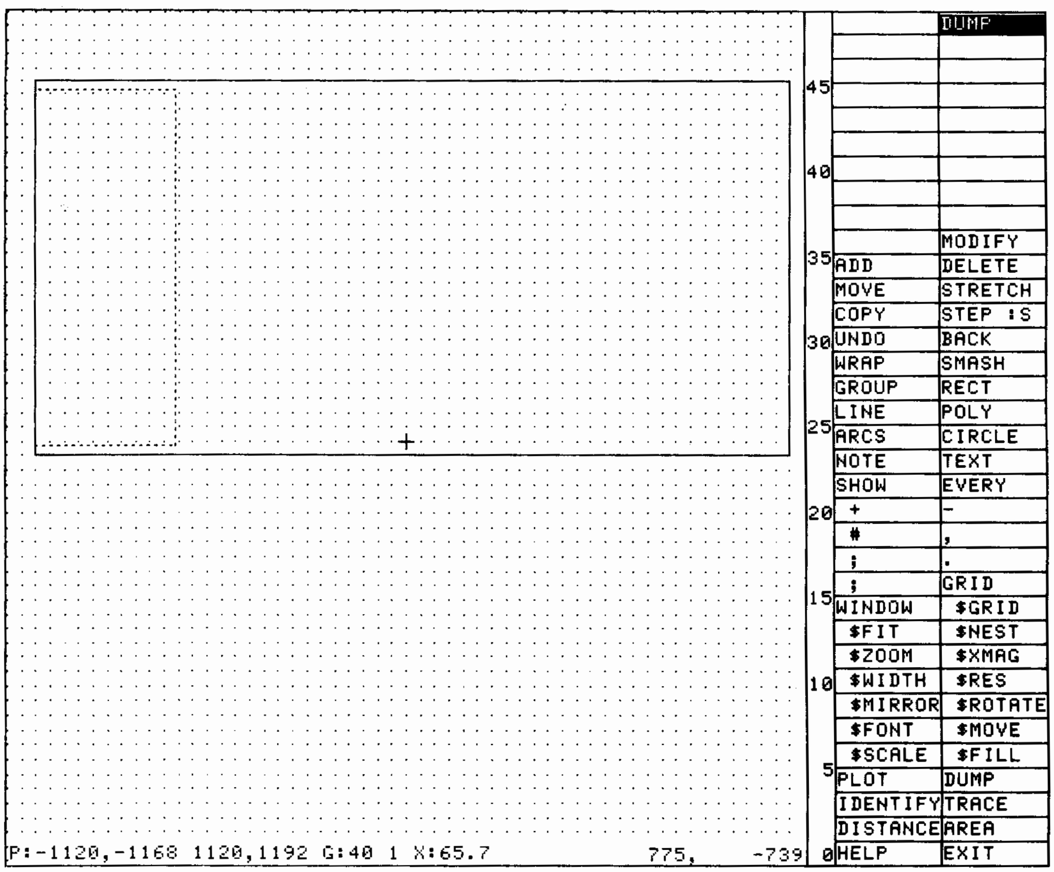


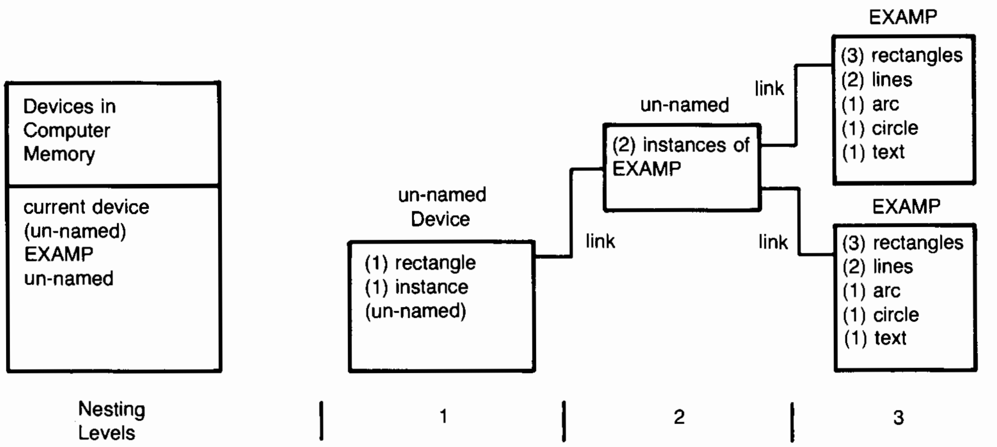
11. Create a new device that consists of the two office plans in the drawing by entering:

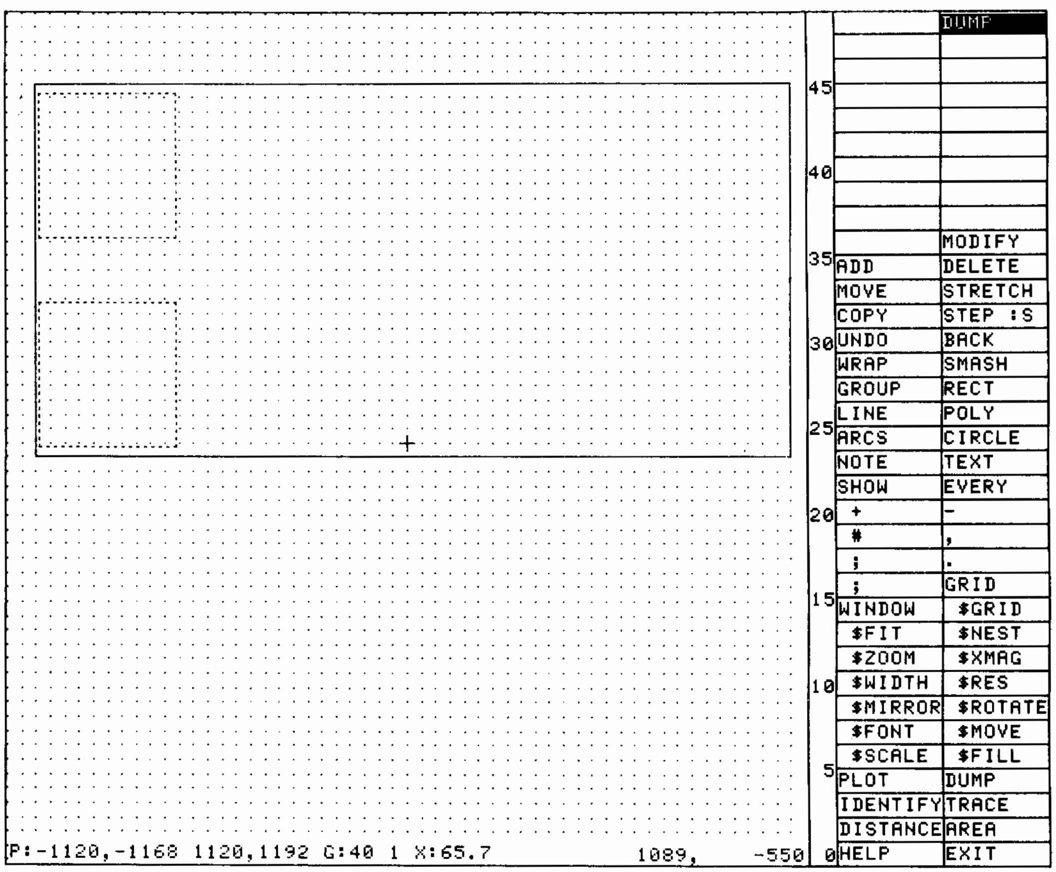
```
WRAP -1040,0 -1080,-80 -560,1040 ;
```

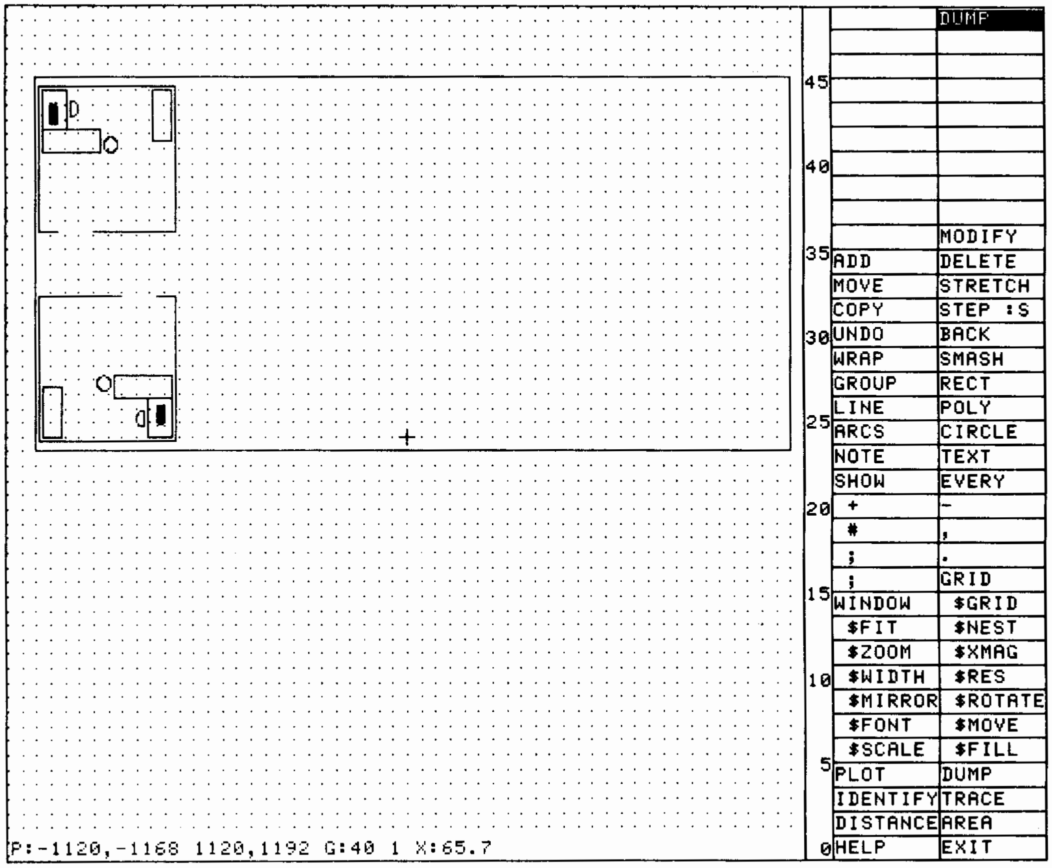
This causes all components completely enclosed by the wrapping rectangle to be defined as a new, unnamed device. Only two components form the new device, two instances of (links to) EXAMP. When the device is created, the instance components in nesting level 1 (the components forming the device) are replaced by an instance of the un-named device. The instances of EXAMP are displayed in nesting level 2. The components of the devices (EXAMP) to which they link are now displayed in nesting level 3. This is illustrated by entering the following commands while refering to the drawing below:

```
WINDOW $NEST 1 ;

WINDOW $NEST 2 ;

WINDOW $NEST 3 ;
```




Devices in
Computer
Memory
current device
(un-named)
EXAMP
un-named
un-named
Device
(1) rectangle
(1) instance
(un-named)
link
un-named
(2) instances of
EXAMP
link
link
EXAMP
(3) rectangles
(2) lines
(1) arc
(1) circle
(1) text
EXAMP
(3) rectangles
(2) lines
(1) arc
(1) circle
(1) text
Nesting
Levels
1
2
3

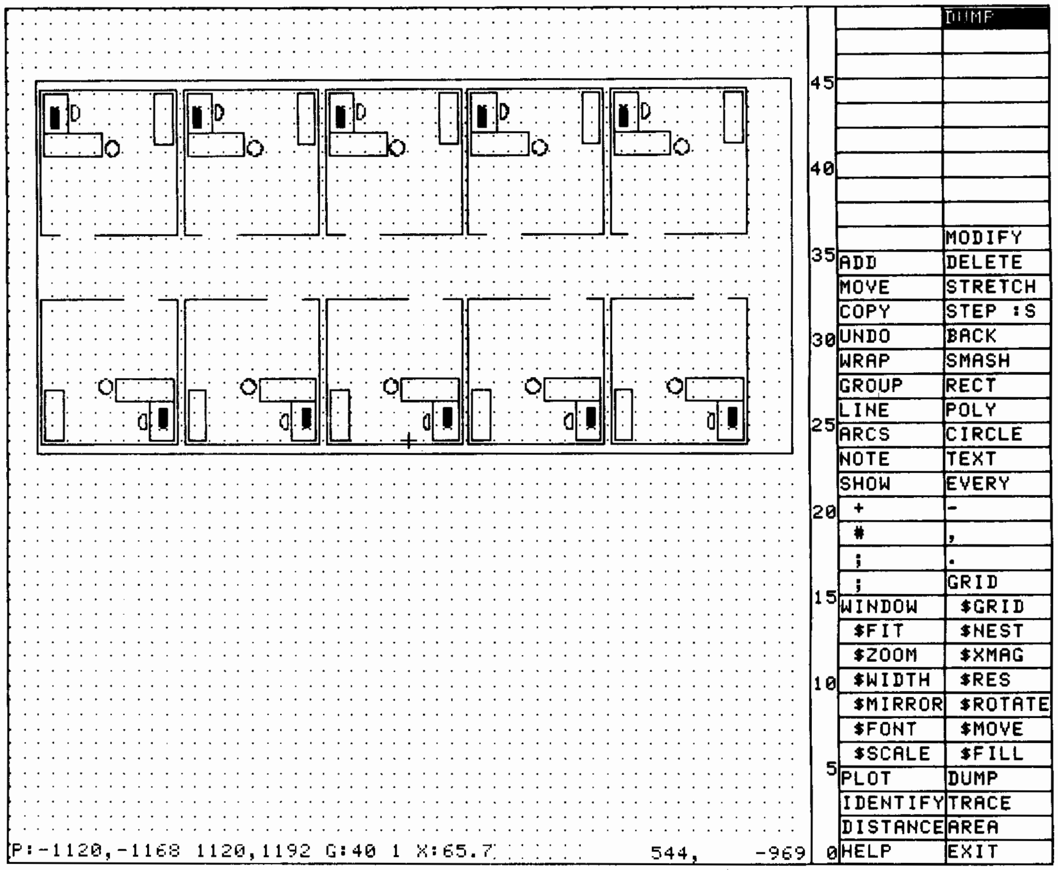

DUMP
MODIFY
ADD
DELETE
MOVE
STRETCH
COPY
STEP :S
UNDO
BACK
WRAP
SMASH
GROUP
RECT
LINE
POLY
ARCS
CIRCLE
NOTE
TEXT
SHOW
EVERY
GRID
WINDOW
$GRID
$FIT
$NEST
$ZOOM
$XMAG
$WIDTH
$RES
$MIRROR
$ROTATE
$FONT
$MOVE
$SCALE
$FILL
PLOT
DUMP
IDENTIFY
TRACE
DISTANCE
AREA
HELP
EXIT
P:-1120,-1168 1120,1192 G:40 1 X:65.7
1089, -550



12. To complete the floor plan, you need eight more office plans placed in the drawing. You could add these by adding more instances of EXAMP, or by using the COPY command to copy individual instances of EXAMP. By making use of the STEP :S command, you can create a 5 (column) by 1 (row) matrix of instances of the newly created, unnamed device. This will add the remaining 8 office plans while only one command is entered. Enter:

```
STEP :S 5,1 -840,480 -1040,0 -600,0 -640,0 ;
```

The first xy location (−840,480) identifies the component to be copied. The second xy location (−1040,0) specifies a reference point for the component to be copied. The third xy location (−600,0) specifies the placement of the copies. This point is a trial placement location. Thus, entering the fourth point replaces the previous placement location with the new placement location (−640,0).

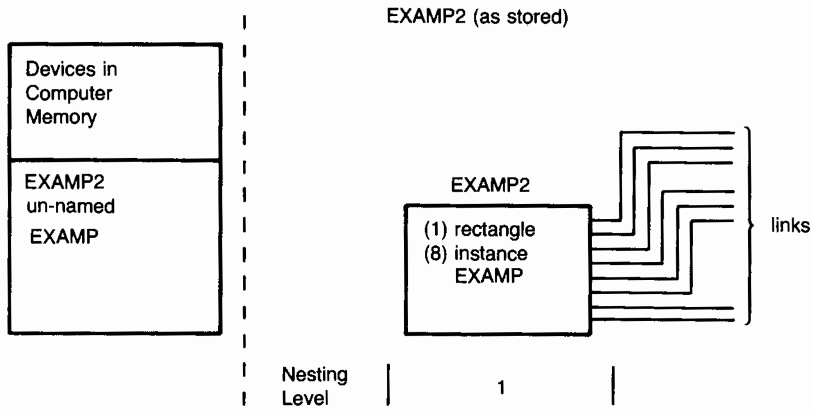


13. Save the new floor plan with the device name, EXAMP2 by entering:

```
SAVE EXAMP2 ;
```

When the drawing is saved, links to unnamed devices are smashed. The components of these devices replace the instance components in the nesting level containing the instances. The drawing below shows the representation of the saved device, EXAMP2.

Only components of EXAMP2 (those displayed in nesting level 1) are stored in the device file, EXAMP2. The components of the linked device (EXAMP) are stored in the device file EXAMP, which you created in Chapter 1.



14. Make a copy of your drawing by entering:

    DUMP ;

15. Exit the edit subsystem by entering:

    EXIT ;

16. Exit the Graphics Editor by entering:

    BYE

Screen Control Commands

The screen control commands allow you to specify: the location of the window over the drawing area, the current user grid, the contents of one or more menu slots, the lock angle, and the scale of a drawing being digitized from the graphic input device.

The function of each of the screen control commands is discussed in the sections below. An interactive example, showing the use of these commands, is provided at the end of the screen control commands.

WINDOW command

The WINDOW command allows you to control the position of the drawing area window over the entire drawing area. Additionally, it allows you to determine the magnification of the drawing viewed in the drawing area window. The WINDOW command also allows you to specify the type layers that are displayed (those layers defined as interconnect, symbolic, or detail layers), which nesting levels are displayed, and whether the user grid is displayed. Any time that the WINDOW command is entered, the drawing area of the display is redrawn.

Changing the location of the drawing area window over the entire drawing area without changing the magnification of the drawing is called panning. By specifying one x,y location with the WINDOW command, the drawing area window is shifted such that the x,y location entered becomes the center of the screen.

Entering two x,y locations with the WINDOW command causes the system to change the location of the drawing area window and the magnification of the drawing. It is changed such that the the lower left corner of the rectangle specified by the two x,y points becomes the x,y location of the lower left corner of the window. Since the ratio of the height to width of the drawing area must remain 453/430, the system calculates the location of the upper right corner of the window such that ratio of height to width is maintained while the entire area defined by the two x,y locations is contained in the window.

For example:


```
WINDOW 0,0 200,200 ;
```

These two x,y points define a rectangle whose diagonal has endpoints on 0,0 and 200,200. Since 0,0 marks the lower left corner of the rectangle, the window is adjusted such that the lower left corner of the window is located at 0,0.

Since you know that the 453/430 ratio must be maintained and since the x and the y differences between points one and two are the same, let the shorter edge of the window (the width) equal 200. The y value of the second point must then be adjusted so that the ratio is maintained. 453/430 x 200 yields the y value, 210.70. Thus, the system sets the window corners at 0,0 and 200, 210.70; the area enclosed by the original two points is included in the drawing area and the ratio of the height to width of the window is maintained. EGS-45 uses this method whenever the window location or magnification of the drawing viewed through the window changes.

Using the zoom option of the WINDOW command, causes the corner values of the window to be divided by the zoom factor (when the zoom factor is positive) and multiplied by the absolute value of the zoom factor (when the zoom factor is negative). This changes the magnification of the drawing displayed in the drawing area, but does not change the x,y location of the center of the display.

The grid option of the WINDOW command, allows you to display/not display the user grid, when the screen is redrawn.

Specifying a nesting level with the WINDOW command causes the system to display the components of all devices that are displayed in nesting levels 1 through the specified nesting depth. Instance components contained in devices at the specified nesting depth are displayed as rectangles in layer 0.

GRID command

The GRID command allows you to define the user grid, which is a subset of the system grid. The UNITS command in the process subsystem assigns user units (such as inches or centimeters) to the internal grid. With the GRID command, you specify the number of user units between user defined grid points. Since units are assigned to the system grid, this also identifies the number of system grid points between user grid points. The user grid determines what locations may be specified in the edit subsystem. Locations can only be entered on a user grid point; attempting to enter a point at a location other than a user grid point causes the point to be snapped (shifted) to the nearest user grid point.

Displaying the entire user grid can cause the screen to become cluttered with grid points. The GRID command allows you to limit the number of user grid points that are displayed. All user grid points are active, but only a portion of them are displayed (for example, display every other user grid point or display every 5th user grid point).

Entering an x,y location with the GRID command, causes the grid to be shifted such that a grid point lies on the location specified. Since all points entered are snapped to the nearest user grid point, it may be necessary to specify an extremely fine user grid before attempting to align the new user grid.

The grid is aligned with the location 0,0 if no alignment point is entered with the command.

LOCK command

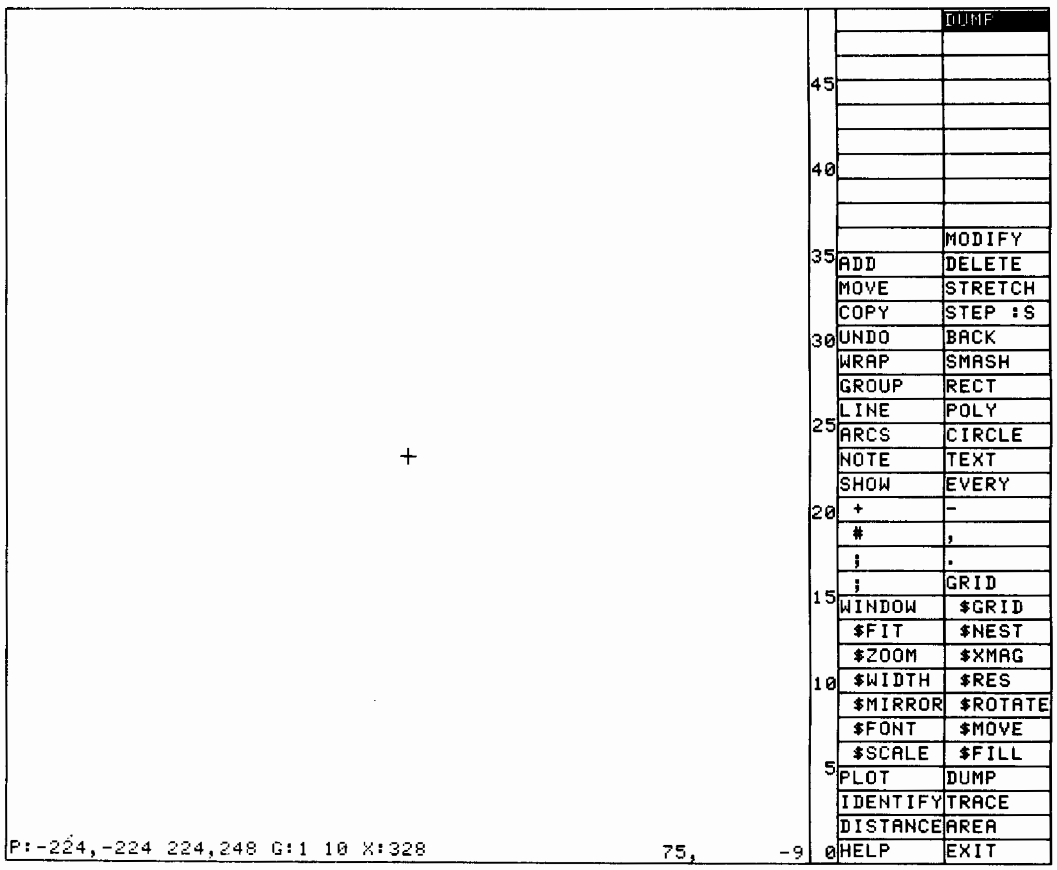
The LOCK command allows you to display or set the system lock angle. The system lock angle is used to determine the default resolution of circles and arcs added to the drawing. The lock angle is also used by the system to flag any component or portion of a component that is not rotated at an integer multiple of the lock angle.

When the angle of a component or portion of a component is not an integer multiple of the lock angle, a message is displayed in the information display area: Lock angle error. This message is not critical, but is intended to aid you when entering components. For example, when laying out traces for a printed circuit board, it is important that the angle formed by lines representing traces does not exceed 45 degrees. Setting the lock angle to 45 degrees would cause the computer to issue a message when ever the angle formed by a component or portion of a component is not an integer multiple of 45 degrees (0, 45, 90, 135 ...).

MENU command

The MENU command allows you to change the contents of an individual menu slot or change the contents of all menu slots by loading a different menu file. Additionally, the menu command allows you to store the contents of the menu in a menu file, thus allowing you to recall that menu for a particular application or purpose.

The menu file, MENUDATA is supplied with EGS-45. It contains the names of several commonly used edit subsystem commands. Additionally, the names of several macros supplied in the macro file, MACRDATA are included in the menu file. These macros replace the single character component descriptors and several of the more cryptic command options. For a list of these macro definitions, see the DEFINE command in the macro subsystem section of this chapter. The display below shows the menu items contained in the menu file, MENUDATA.



SHOW command

The SHOW command allows you to specify which of the system's 255 layers are active and may be drawn on and which components may be placed in those layers. Before a layer can be turned on (activated) with the SHOW command, it must be defined in the process file or with the EQUATE command in the process subsystem.

When you enter the edit subsystem, all components in layers are displayed and are modifiable. The effects of the SHOW command are cumulative. For example, if you first specify to turn off all components in all layers (SHOW -E), then specify to turn on text components in all layers (SHOW #T), and finally specify to turn on line components in layers 2 and 3 (SHOW #L2,3), the net effect is to have all text components in all defined layers displayed along with line components in layers 2 and 3; no other components are displayed.

Three options are available with the SHOW command. You may specify that components in a layer are not to be displayed at all, that components in a layer are displayed but cannot be modified, or that components in a layer are displayed and can be modified.

The SHOW command can be used to temporarily remove unnecessary items from the drawing. For example, if you are creating a drawing with many notes, it might be convenient to "turn off" the note components to decrease the amount of time needed to redraw the screen by entering SHOW -N. This causes a note component contained in any layer to not be displayed. The note component is not lost or destroyed, but can be re-displayed at any time by turning on the note component with the SHOW command.

CALIBRAT command

The CALIBRAT command allows you to digitize a diagram or sketch from the graphic input device and add it to the drawing area of the display. The CALIBRAT command allows you to assign a scale to the diagram, so that the units of the diagram match the units of the drawing. It also allows you to correct for the angle at which the diagram is placed on the graphic input platen.

You set the scale of the diagram by entering two x,y pairs (usually by typing in these values) that represent the x and y distance between edges of the drawing. For example, entering the values 0,20 and 100,220 assigns a scale of 100 user units (100 - 0) to the length of the diagram and a scale of 200 user units (220 - 20) to the height of the diagram.

Next, four x,y locations are digitized from the graphic input device. These four points should be the four corner locations of the rectangular area encompassing the diagram (all four points must be within the active platen area of the graphic input device). From the location of these points, the system determines the angle of rotation of the diagram on the platen and the location of the edges of the diagram.

The drawing area window is adjusted such that the lower left corner of the diagram "maps" to the lower left corner of the drawing area while the area defined by the four points is mapped into the drawing area. The magnification of the drawing through the window (the zoom) is adjusted such that the scaled diagram, defined by its four corners, is just contained within the drawing area. Because of the requirement that the ratio of the height to width of the drawing area remain 453/430, some extra drawing area may be shown at the right or top of the display.

Changing the location of the window or the magnification of the drawing viewed in the window (the zoom), disables the CALIBRAT command, thus allowing normal interaction with the system (no scale or rotation considered when selecting points on the input device).

If the CALIBRAT command is entered and no scale or corner locations are entered with it, the system re-enables the scale of the last CALIBRAT command entered and the mapping of the area defined by the last four corner parameters is again enabled. However, the window of the drawing area is not changed and the WINDOW command must be used to shift the window to the desired location. If the location of the window is not shifted, the portion of the total drawing area that maps to the graphic input platen may not be in the window. Thus, you will not be able to digitize the sketch.

Sketches created with lead pencil or other electrical/magnetic conductive materials cannot be digitized from either the HP 9111 graphics tablet or the HP 9874 digitizer. For a complete list of problem materials, see the hardware manual associated with that device.

A Simple Example

Before discussing the capabilities of the other edit subsystem commands, a simple interactive example is presented. By working along with this example, you will gain a better understanding of the concepts of the system. Additionally, by following the syntax diagram for each command (located in the Syntax Reference), you can gain an understanding of the different options offered by each command.

You will edit the office floor plan drawing that you created in the previous interactive example, EXAMP2. You will use the screen control commands to change the way in which this drawing is displayed. Since the concept of nesting levels was illustrated heavily in the previous example, the subject is not discussed in this example.

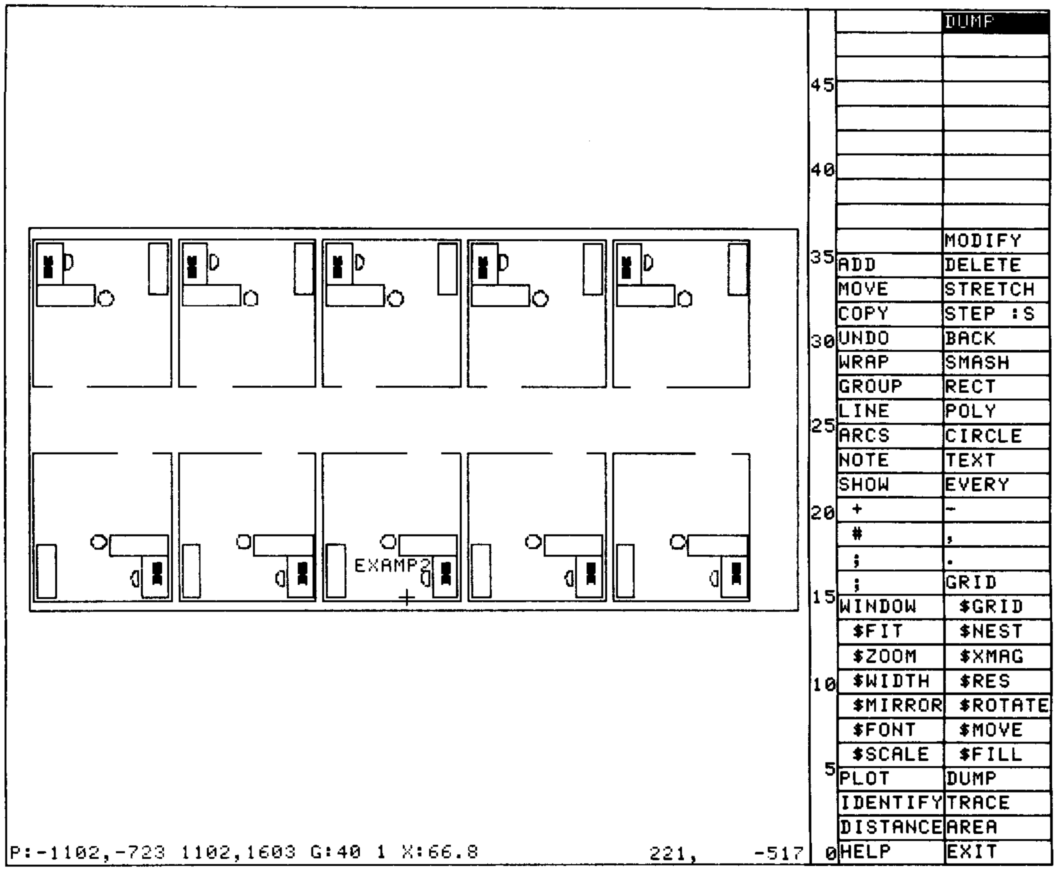
As you work the example, use the technique of selecting to enter commands, parameters, and data from the menu are of the display. When this is not possible, use the computer keyboard to enter the information.

Perform the following steps.

1. Load and run the EGS-45 software.
2. Select the Graphics Editor from the menu on the CRT.
3. List the contents of the current process file by entering:
   ```
   LIST PRO ;
   ```
4. Enter the edit subsystem to edit the device EXAMP2 by entering:
   ```
   EDIT EXAMP2 ;
   ```
5. Display all components of EXAMP2 and all components of linked devices by entering:
   ```
   WINDOW $NEST 3;
   ```
   This causes all components of all devices in nesting levels 1 through 3 to be displayed.

6. Shift the location of the window (pan over) such that the xy location 0,440 becomes the center of the window. Enter:

   WINDOW 0,440 ;



7. Use the SHOW command to "turn off" the display of every component in layer 2 by entering:

   SHOW -EVERY 2 ;

8. Turn on the display of all components again by entering:

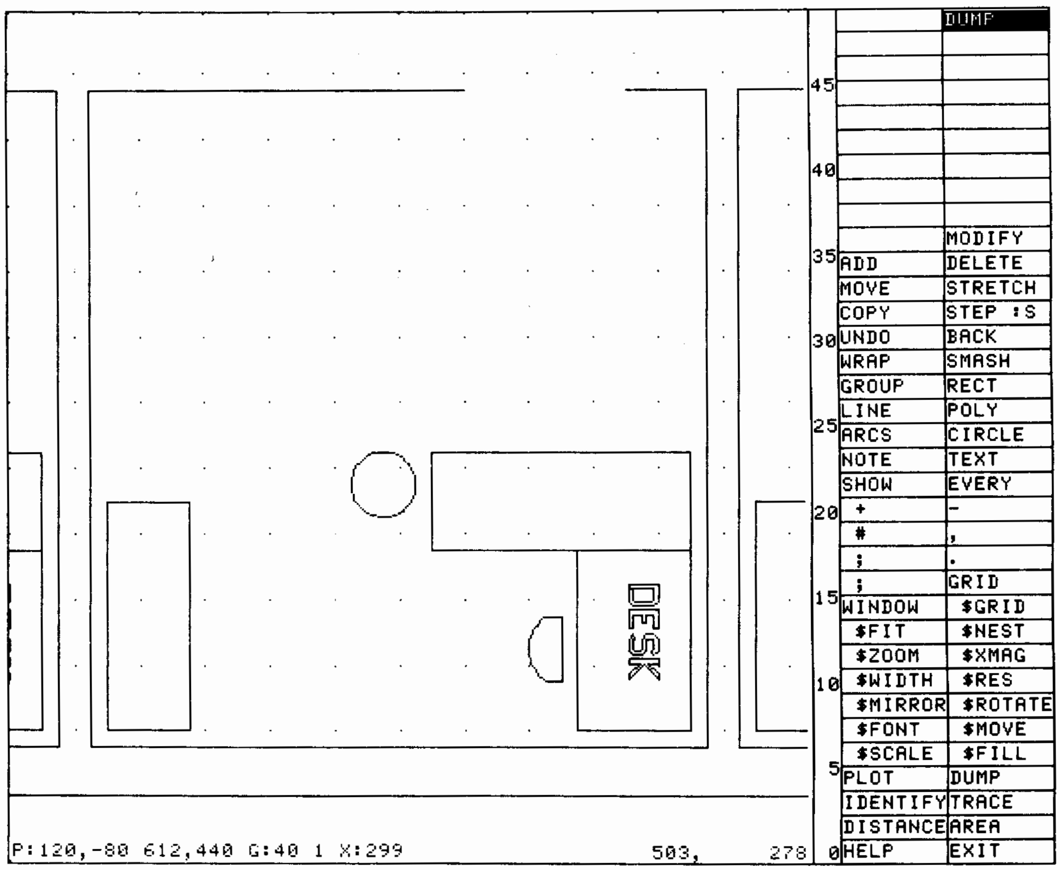
   SHOW #EVERY ;

9. Turn on the display grid. The display grid is equal to the user grid since every user grid point is displayed by entering:

   GRID ;

10. Take a close up look at one of the offices by entering:

    `WINDOW 120,-80 600,440 ;`

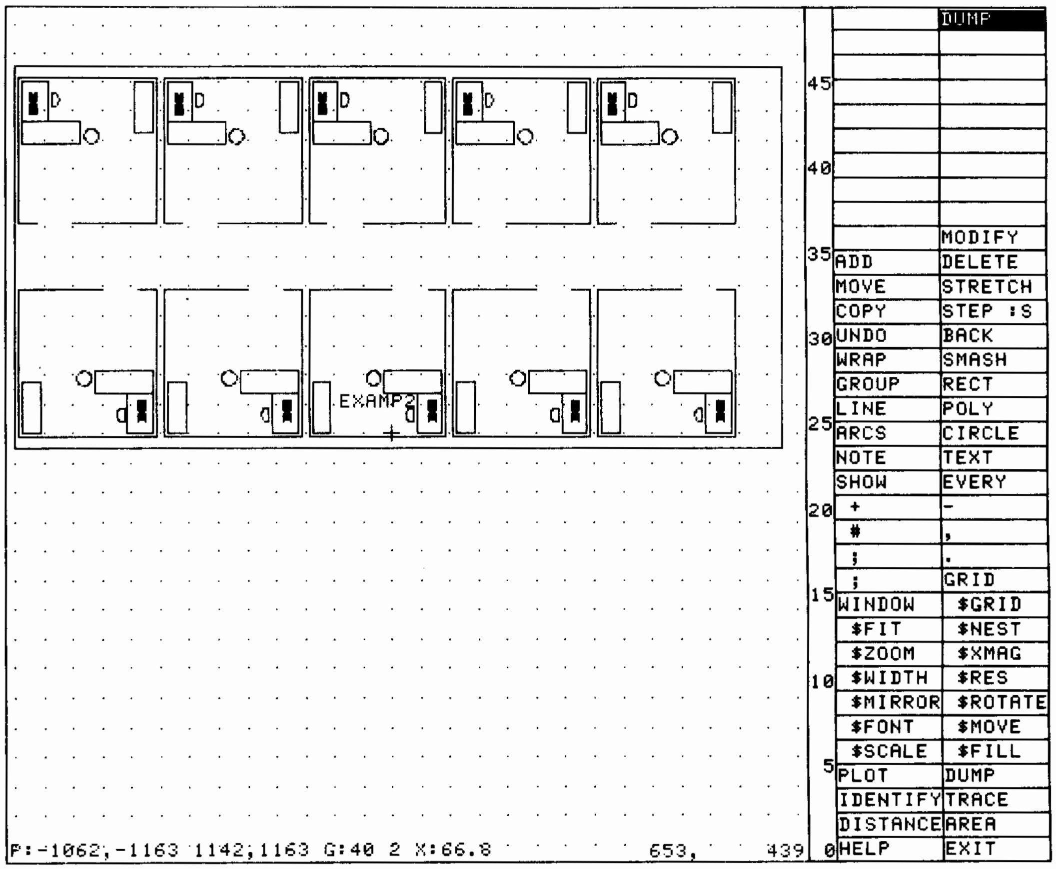


11. Display the entire office plan in the window and set the center of the window to be 0,440 by entering:

    `WINDOW $FIT; 0,440 ;`

12. The current user grid is set to 40 user units per user grid point. The current display grid is set to display every single user grid point. Change the display grid to display every other (every 2nd) user grid point by entering:

    GRID 40,2 ;



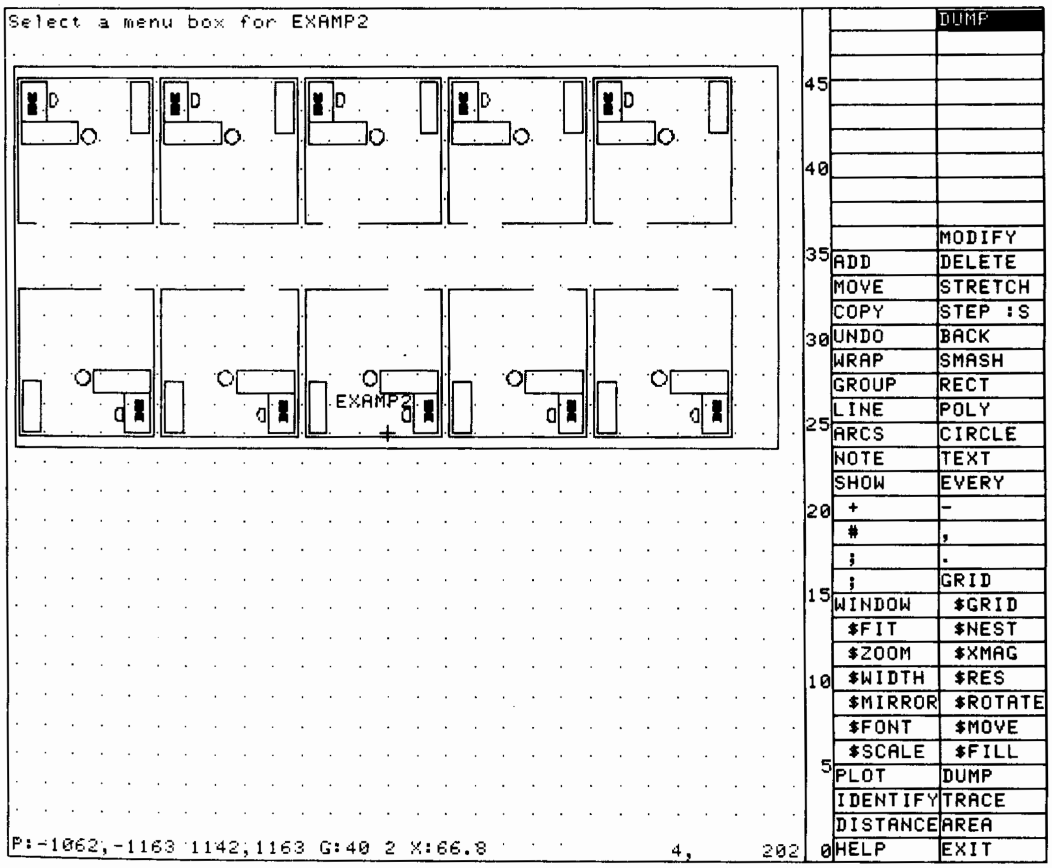
13. Redraw the screen by entering:

    WINDOW ;

14. Use the MENU command to add the characters EXAMP2 to the menu. This allows you to add instances of the device EXAMP2 by selecting the ADD command and the menu box containing the device name, EXAMP2. Since several of the characters for this command are not available in the menu, you will need to type these characters on the computer keyboard. Enter:

```
MENU 'EXAMP2'
```

and select any vacant menu slot, except the menu slot at the top right (this is reserved by the system for displaying the current command).



15. Since no modifications were made to the drawing, you will exit the edit subsystem without saving the drawing. Enter:

```
EXIT ;

EXIT ;
```

16. Exit the Graphics Editor by entering:

```
BYE
```

Modification Commands

The modification commands are used to change or modify components and drawings, once they have been added or created. They allow you to: move a component from one location in the drawing to another, remove a component from the drawing, modify the shape of some components (arcs, circles, lines, polygons, and rectangles), correct mistakes made during the entry process, and smash an instance component (replaces the link to the device with the components that form the device).

Backup command (∧)

The backup command is designed to be used in conjunction with edit subsystem commands that require x,y data be entered. The backup command (∧) allows you to delete, or back over, an x,y point that has been entered with a command. The backup command is valid only if the command has not been terminated. X,y locations are deleted in LIFO (last in, first out) fashion; hence, the term backup. For example, `ADD LINE 1 xy1 xy2 xy3 xy4 ^ ^` causes the last two points entered with the ADD command to be deleted. Since the command has not been terminated, you may enter new x,y points to replace the ones that you have deleted. The backup command is in effect, an electronic eraser that allows you to correct mistakes as they are made, instead of deleting the entry and starting over.

Data points cannot be erased with the backup command once a command has been terminated.

DELETE command

The DELETE command allows you to delete a component from the drawing. Specifying an x,y location that lies in or on the component with the DELETE command selects that component as the component to be deleted. However, the component is not deleted until the command is terminated, thus allowing you to re-select the component to be deleted if an error has been made. An optional descriptor and layer number may be entered to limit selection of the component.

MODIFY command

The MODIFY command allows you to modify any existing component on the screen. The types of modification which can be performed include rotation, mirroring, scaling, changing layer numbers, changing font size of text and notes, changing width of lines, and moving the component. You may perform all or any combination of these modifications with one command.

MOVE command

The MOVE command allows you to move any component from its current location to a new location on the same layer. The component to be moved is selected by specifying a location that lies in or on the component. A reference point is assigned to the component, allowing you to accurately place the component at its new location. The component is moved such that the reference point coincides with the destination location. An optional descriptor and layer number may be entered to limit selection of the component to be moved.

SMASH command

The SMASH command allows you to replace an instance component (link to a device) with the components of the device to which it links. When an instance is smashed, the instance component is removed from the drawing and the components that form the linked device are added to your drawing. These components are then displayed in the same layers as they were added in the linked device.

STRETCH command

The STRETCH command allows you to change the shape of a component by moving the location of an endpoint, a vertex, a radius, or a side. Only arcs, circles, lines, polygons, and rectangles may be stretched. To stretch a component, you first select the component that you wish to modify. Next, by entering an xy point, you specify the part of the component (side, vertex, ...) that is to be moved and assign a reference point. Finally, you specify a destination location for the part of the component and that part of the component is moved such that the reference point lies on the destination location. Specifying additional locations causes the part of the component to be moved to the new location, thus allowing you to experiment and find the exact location where you want the part to be positioned.

UNDO command

The UNDO command cancels the previously entered command. For example, if the previous command was to delete a rectangle, entering the UNDO command causes the rectangle to be restored to the drawing, thus "undoing" the effects of the DELETE command. The UNDO command can reverse only the effects of the last command entered. Only the following commands can be "undone": ADD, COPY, DELETE, GROUP, MODIFY, MOVE, SMASH, STEP :S, STRETCH, and WRAP.

A Simple Example

Before discussing the capabilities of the remaining edit subsystem commands, a simple interactive example is presented. By working along with this example, you will gain a better understanding of the concepts of the system. Additionally, by following the syntax diagram for each command (located in the Syntax Reference) as you enter it, you can gain an understanding of the different options offered by each command.

In this example, you will modify the office complex floor plan EXAMP2 that you created earlier in this chapter. You will use the drawing modification commands to alter this drawing and to demonstrate the use of the different commands.

Use the technique of selecting to enter commands, parameters, and data from the menu area of the display, when possible. Use the keyboard to enter commands only when the information to be entered is not contained in the menu. This allows you to practice with the selection technique.

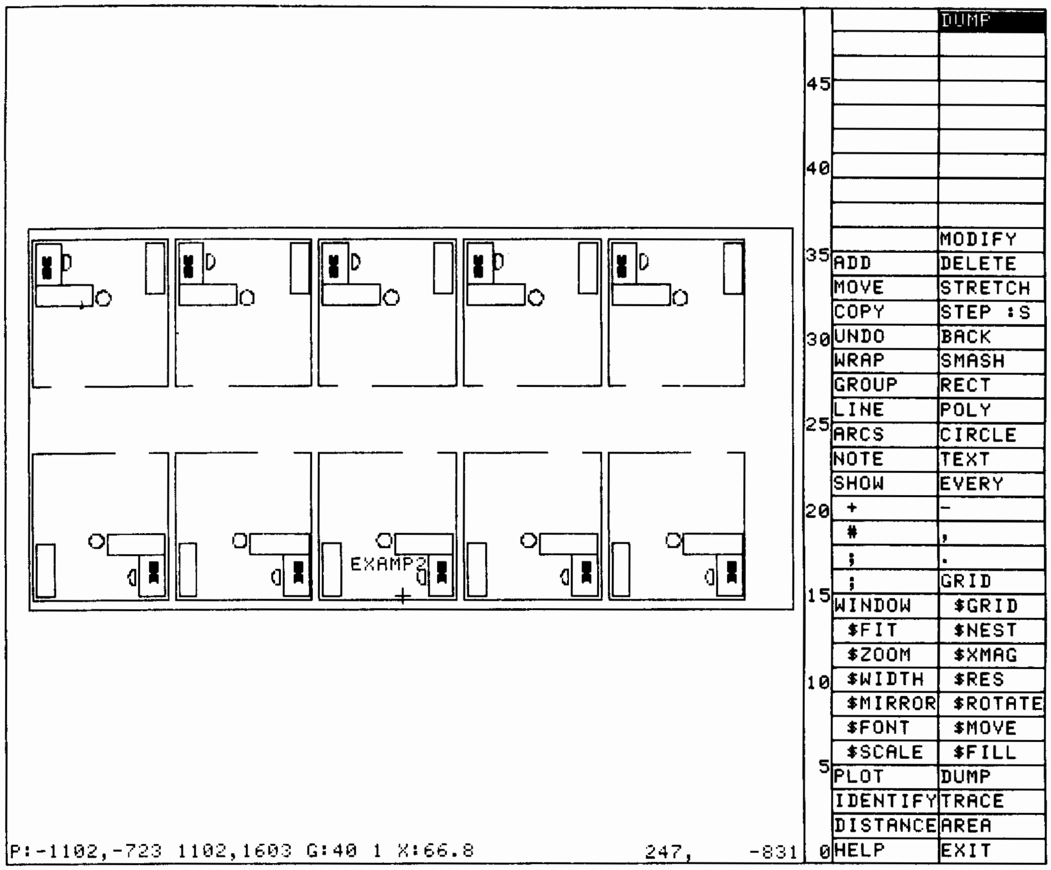
Perform the following steps.

1. Load and run the EGS-45 software.
2. Select the Graphics Editor from the menu on the CRT.
3. Enter the edit subsystem to edit the device EXAMP2 by entering:

```
EDIT EXAMP2 ;
```

4. Display all components in nesting levels 1 and 2 and place the center of the window over the location 0,440 by entering:

```
WINDOW $NEST 2 0,440 ;
```



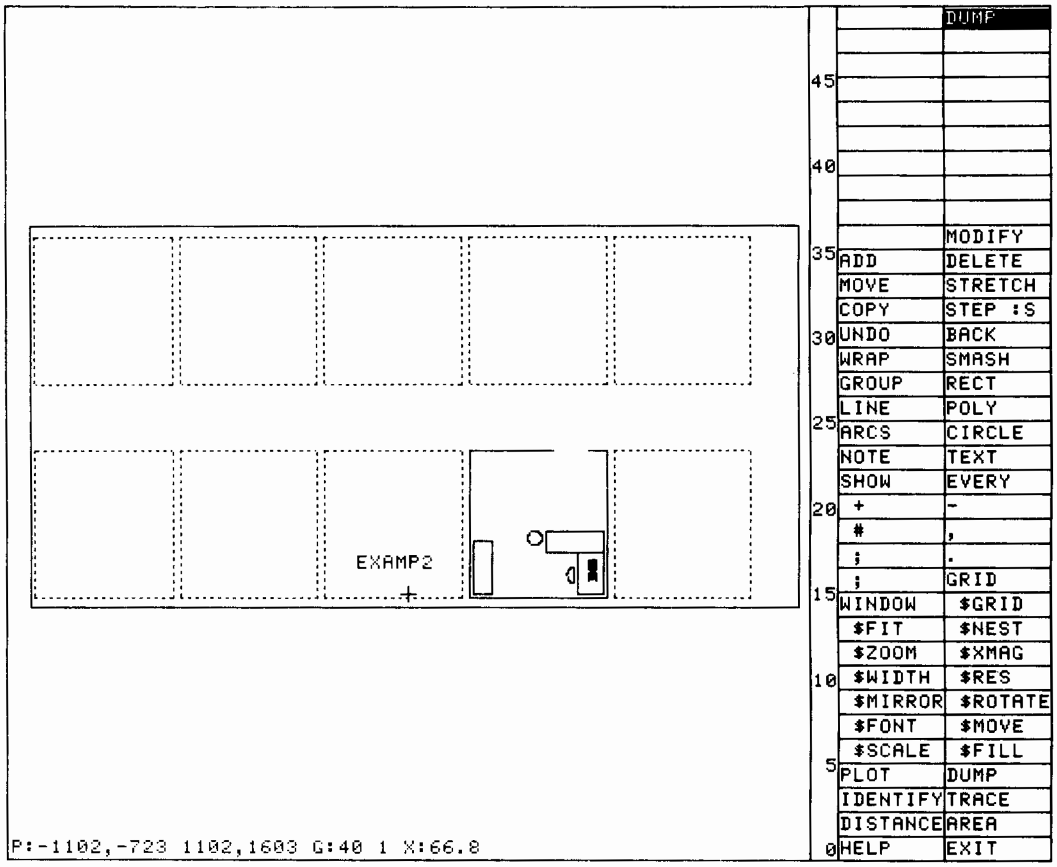
5. The commands of the edit subsystem can act only on components in nesting level 1, regardless of the number of nesting levels displayed. Thus, in the drawing you cannot access the individual components of the offices (such as the desk). To remove the wing on a desk in one of the offices, you must first smash the link to the device. When the link to a device is smashed, the link is removed from nesting label 1 and is replaced by the components that form the linked device. To illustrate, first display the only components in nesting level 2 by entering:

```
WINDOW $NEST 2 ;
```

6. Now, smash an instance component by entering:

```
SMASH 440,120 ;
```

where 440,120 is an xy location lying in or on the instance component.



7. Take a close look at the office by entering:

```
WINDOW 120,-120 600,440 ;
```

8. Now, use the DELETE command to remove the wing of the desk:

```
DELETE 440,120 ;
```

9. Set the user grid such that there are 10 user units per user grid point. Display every single user grid point. Enter:

```
GRID 10,1 ;
```

Turn the grid on by entering:

```
GRID ;
```

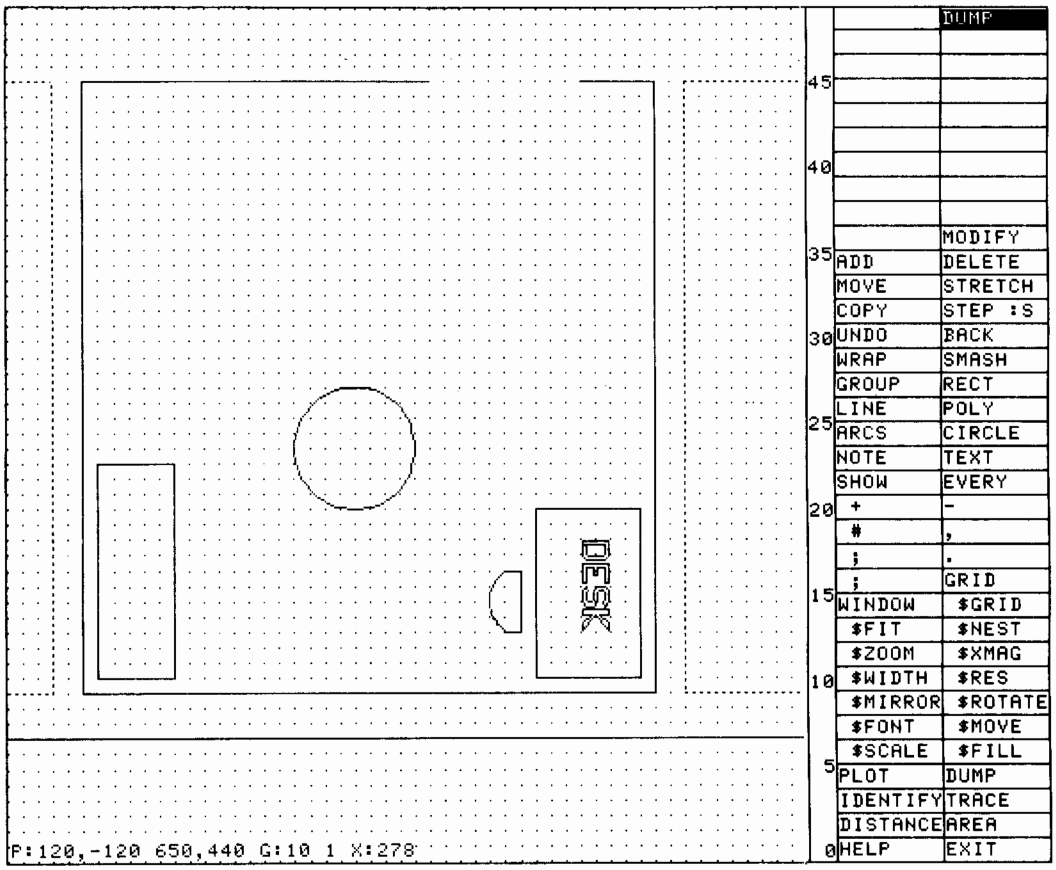
and re-draw the screen by entering:

```
WINDOW ;
```

10. Increase the size of the wastebasket in the office by stretching it. Enter:

```
STRETCH 350,150 370,150 390,150 ;
```

The first xy location entered (350,150) identifies the component to be stretched. The second xy location (370,150) specifies the axis that you wish to stretch. The third xy location (390,150) specifies the increase in length (390-370).



11. Experiment with the drawing modification commands. Use the syntax diagrams in the Syntax Reference to aid you. When you are finished, exit the edit subsystem (without saving the modified drawing) by entering:

```
EXIT ;

EXIT ;
```

Information Commands

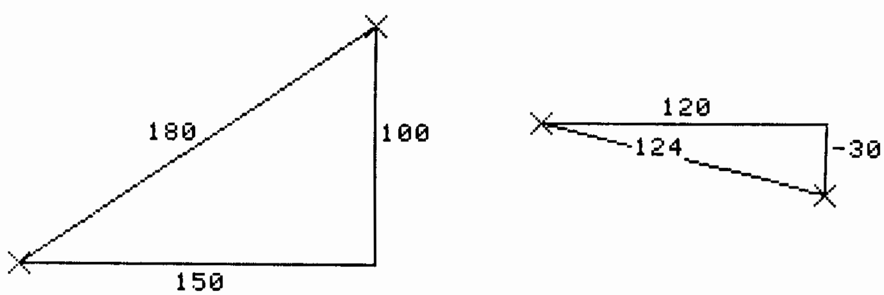
The information commands are commands that provide information about the drawing, the drawing area, or the commands of the edit subsystem. They allow you to: find the distance between x,y locations, find the area enclosed by polygons, display and/or change the inames and tnames assigned to components, display the x,y coordinates of location, and provide a syntax help message for any command in the edit subsystem.

AREA command

The AREA command allows you to display the area, in current user units, enclosed by a circle, a line (with width), a polygon, and a rectangle and the accumulated area of one or more of these components contained in the drawing area of the display. Specifying a location that lies in or on a component causes the area enclosed by that component to be displayed with the label, `AREA=`. The total area of all components selected since the AREA command was entered is displayed with the label `CUM=`. Entering the AREA command again resets the accumulated area to 0.

DISTANCE command

The DISTANCE command allows you to display the x distance, the y distance, and the net distance between two xy locations. This information is erased when the screen is redrawn.



ECHO command

The ECHO command in the edit subsystem works in the same fashion as the ECHO command in the Graphics Editor. It allows you to enable/disable echo. When echo is enabled, each command entered is printed on the current system printer. This provides a log of the commands entered which makes it easy to find an erroneous entry.

When echo is disabled, commands are only displayed on the CRT as they are entered (commands entered via the graphic input device are displayed in the alphanumerics portion of the CRT, the portion you see when typing commands in the edit subsystem).

HELP command

The HELP command allows you to obtain a quick syntax reference for any or all commands valid in the edit subsystem. If syntax help is sought for only one command, the information is shown at the top of the display. If syntax help is sought for all commands in the edit subsystem, the information is printed on the system's internal printer.

For example:

```
HELP WINDOW ;

WINdow [{:S|:I|:D}] [:G] [:F] [:Nlevel] [:Z[pwr]] [:Xmag]
[xy1[xy2]] EOC
```

The command name is displayed showing the first three letters in capital letters while the remainder of the command name is shown in lower case. This indicates that only the first three letters of the command name need be entered.

Items enclosed in square brackets [ ] represent optional parameters that may be entered with the command. Items represented in lower case within the brackets are short notes indicating the type of data or parameter expected. Colons (:) followed by a capital letter are characters indicating an option and must be entered exactly (for example, :G or :F).

Items enclosed in braces { } represent a choice of items, where at least one of the choices must be entered. Choices are separated by a bar, |. For example, in the command above, {:I|:S|:D} indicates that either :I, :S or :D must be entered.

These help messages are intended to be a quick reference for the experienced user. The help message for each command is listed with the syntax diagrams in the syntax reference chapter of this manual. In this way you can quickly learn to associate the short parameter descriptions with the longer explanations provided with the syntax diagrams.

IDENTIFY command

The IDENTIFY command allows you to display the iname assigned to a component, assign an iname to a component, and move the location of the iname. The iname is a string of characters that is associated with a particular component. It functions as a note that stays with the component wherever it is located in the drawing.

The IDENTIFY command allows you to cause all components with a particular iname to be highlighted in the display. Additionally, it allows you to display the iname assigned to a particular component, and to move the location of the iname assigned to the component.

The iname gets its name from the fact that the IDENTIFY command is used to display the name associated with a component.

LIST command

The LIST command allows you to get a listing identifying the macro file, menu file, and process file currently into the system.

You may also obtain a listing describing any device present in computer memory. This listing provides a list of all devices linked to your drawing and a list of the number of each type of components used to create the current drawing. Additionally, it allows you to list the contents of the current process or macro files.

POINT command

The POINT command allows you to display the x,y location of the point specified. This information is only temporary and is erased when the screen is redrawn.

TRACE command

The TRACE command allows you to display the tname assigned to a component, assign a tname to a component, and move the location of the tname. The tname is a string of characters that is associated with a particular component. It functions as a note that stays with the component wherever it is located in the drawing.

The TRACE command allows you to cause all components with a particular tname to be highlighted in the display. Additionally, it allows you to cause the tname assigned to a particular component to be displayed, and to move the location of the tname assigned to the component.

The tname gets its name from the fact that the TRACE command is used to display the name associated with a component.

A Simple Example

This example is designed to introduce you to the information commands. By working along with the example, you will gain a better understanding of concepts of the system. Additionally, by following the syntax diagram for each command (located in the Syntax Reference) as you enter it, you can gain an understanding of the different options offered by each command.

Perform the following steps.

1. Load and run the EGS-45 software.
2. Select the Graphics Editor from the menu portion of the CRT.
3. List the contents of the current process file by entering:

   `LIST PRO ;`

4. Enter the edit subsystem by entering:

   `EDIT ;`

5. Enable echo by entering:

   `ECHO ;`

   This causes every command entered to be printed on the computer's built-in printer. You may disable this option at any time during this example by entering `ECHO ;` again.
6. To obtain a printed listing of the help messages provided by the system, enter:

   `HELP ;`

7. Set the user grid such that there are 10 user units per user grid point. Display every user grid point. Enter:

   ```
   GRID 10,1 ;

   GRID ;
   ```

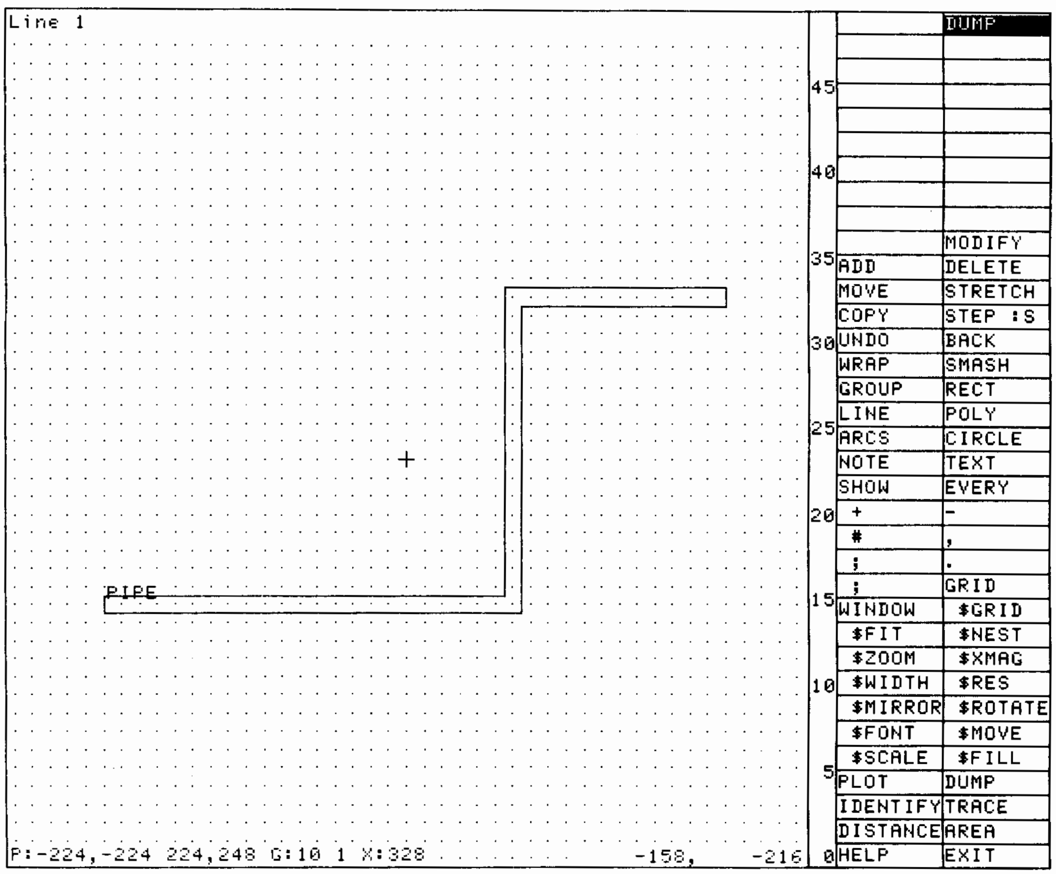
8. Add a line to the drawing area that represents a gasline pipe. Assign the iname, PIPE to the component. Additionally, assign the tname, GAS_LINE to the component. Enter:

   ```
   ADD LINE 1 $WIDTH 10 .PIPE @GAS_LINE -170,-80 60,-80 60,90
   170,90 ;
   ```

9. To verify that the proper iname and tname have been assigned to the line component, enter:

   ```
   IDENTIFY -170,-80 ;
   ```

   This causes the iname assigned to the selected component to be displayed. It is erased when the screen is redrawn with the WINDOW command.



Enter:

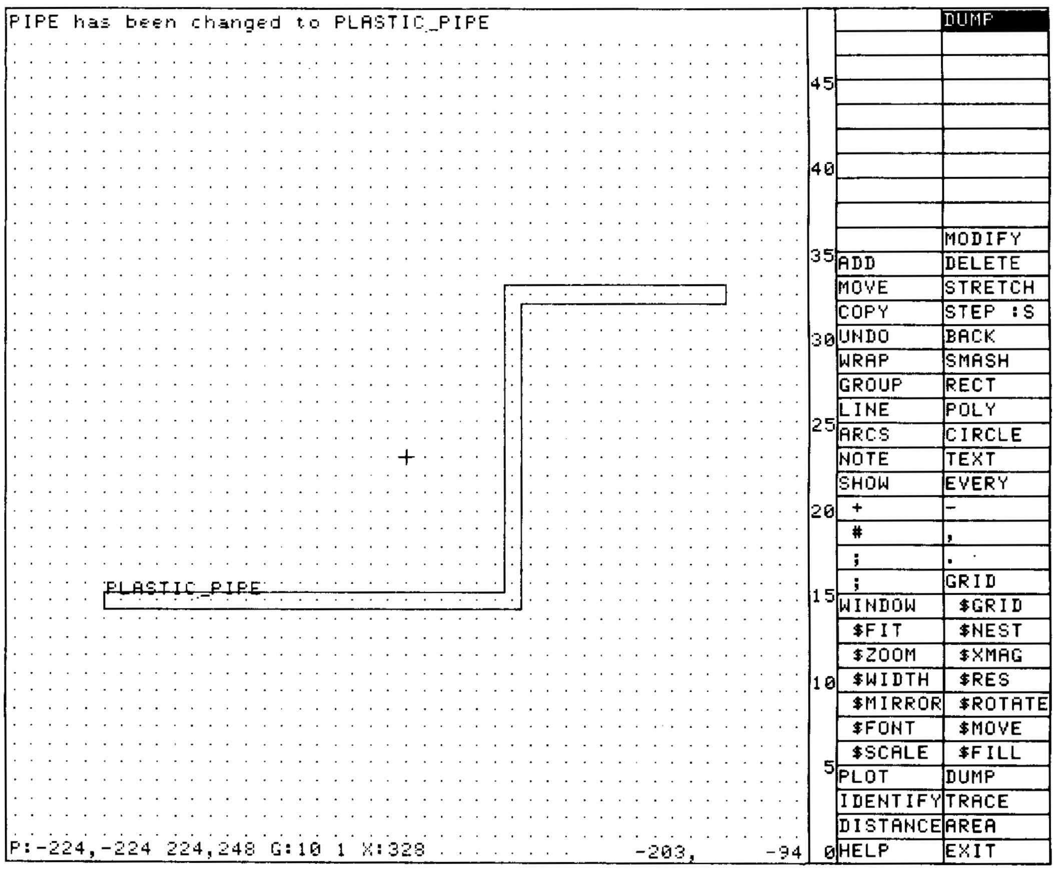
```
TRACE -170,-80 ;
```

This causes the tname assigned to the component to be displayed. It also is erased when the screen is redrawn.

10. Change the iname assigned to the line component from PIPE to PLASTIC_PIPE by entering:

```
IDENTIFY ,PLASTIC_PIPE -170,-80 -170,-80 ;
```

Selecting the component the first time caused the iname assigned to the component to be displayed. Reselecting the component causes the system to change the iname assigned to the component to PLASTIC_PIPE.



11. The DISTANCE command displays the distance between two specified points. Enter:

```
DISTANCE -160,10 -30,90 ;
```

and observe the results. The information displayed is erased when the screen is redrawn.

12. The POINT command displays the xy location of the point specified. Enter:

```
POINT
```

and select one or more points in the display area.

13. Redraw the screen by entering:

    ```
    WINDOW ;
    ```

14. Find the area of the pipe in the drawing area by entering:

    ```
    AREA -170,-80 ;
    ```

15. Experiment with the information commands. When you are finished, exit the edit subsystem without saving the drawing by entering:

    ```
    EXIT ;

    EXIT ;
    ```

Macro Subsystem

Overview

The macro subsystem allows you to create new commands, called macros, from combinations of existing commands. Many macros can be created and stored in an EWE file, called a macro file. Loading a new macro file causes the macros contained in that file to be defined, and replace the former macro definitions that may have been present. Once a macro command is defined, the macro name may be added to a menu slot in the menu area in the edit subsystem, thus allowing the macro to be selected.

The macro subsystem is entered by entering the MACRO command in the Graphics Editor and a specific macro file may be loaded into the system at this time. Some basic concepts are provided to help you understand the macro subsystem, followed by a description of each of the commands available in the macro subsystem.

Concepts

Macro commands

A macro command is simply a new command that is created from one or more existing commands. You can create macros that are valid in the Graphics Editor or any of its subsystems. Macro commands are usually created to speed command entry by combining commands that are often entered together. Other times macros are created to make a command or parameter more intuitively obvious, by simply changing the name of the command or parameter.

Macro Subsystem Commands

DEFINE command

The DEFINE command is the command used to create a macro command. A macro definition involves specifying: the macro name, space holders for data required by the commands forming the macro (including default values for the data), and finally the commands that form the macro. A macro may be defined using other defined macros as well as existing Graphic Editor commands and the commands of its subsystems.

As an example of macros, the macro file MACRDATA is supplied with your EGS-45 system. The macros contained in this file are designed to replace the more cryptic option parameters that accompany commands (such as :R, :MX, etc). Additionally, several macros are provided to replace the single character component descriptors with a word that more fully represents the component. Several of these macro names have been placed in the supplied menu file, MENU-DATA. This allows these macros to be selecting from the menu area of the display in the edit subsystem. A list of the macros supplied in this file is supplied below.

Macro name	Replaces	Description of use
BACK	^	Replaces the character ^ as the entry for the BACKUP command.
$RES	:R	Used to enter the resolution of either a circle or an arc component.
NOTE	N	Macro for the note component descriptor.
TEXT	T	Macro for the text component descriptor.
INSTNCE	I	Macro for the instance component descriptor.
RECT	R	Macro for the rectangle component descriptor.
CIRCLE	C	Macro for the circle component descriptor.
LINE	L	Macro for the line component descriptor.
EVERY	E	Macro for the all-inclusive component descriptor.
POLY	P	Macro for the polygon component descriptor.
ARCS	A	Macro for the arc component descriptor.
$FIT	:F	Used to enter the fit option (:F) with the WINDOW command.
$GRID	:G	Used to enter the grid option (:G) with the WINDOW command.
$ZOOM	:Z	Used to enter the zoom option (:Z) with the WINDOW command.
$XMAG	:X	Used to enter the absolute magnification option with the WINDOW command.
$NEST	:N	Used to specify the maximum nesting level displayed with the WINDOW command.
$WIDTH	:W	Used to enter the width option when adding lines to your drawing with the ADD command.
$MIRROR	:M	Used to specify the mirror option when adding instance components, note components, or text components with the ADD command.
$ROTATE	:R	Used to specify the rotate option with the commands of the edit subsystem that allow rotation.
$MOVE	:M	Used to specify the move option with the IDENTIFY and TRACE commands.
$FONT	:F	Used to specify the fontsize of characters when adding text or note components with the ADD command.
$SCALE	:X	Used to specify the scale option with either the PLOT command or with the ADD command (when adding instace components).
$FILL	:F	Used to specify the fill option with the PLOT command.

Notice that a space precedes most of the characters replaced in the macro definition. The space character allows the macro to be selected from the menu area of the display in the edit subsystem. The space character ensures that a delimiter separates the command and the parameter characters.

Notice also that several macros have the same definition (they replace the same set of characters). For example, both $RES and $ROTATE are used to replace the characters (:R). Since entering a macro simply causes the macro to be replaced with the characters that define it, it makes no difference which macro is entered. To specify an arc component with a resolution of 6 degrees, any of the following entries could be made since they are all equivalent:

```
ADD ARCS 1 $ROTATE 6...

ADD ARCS 1 $RES 6...

ADD ARCS 1 :R 6...
```

ECHO command

The ECHO command allows you to enable/disable echo. When echo is enabled, each command entered is printed on the current system printer. This provides a log of the commands entered which makes it easy to find an erroneous entry.

When echo is disabled, commands are only displayed on the CRT as they are entered.

EXIT command

The EXIT command allows you to exit the macro subsystem and return to the Graphics Editor. If any new macro definitions have been created and not stored in a macro file, the system issues a warning message that the macro file has not been saved. You may continue working in the macro subsystem, save the macro file with the SAVE command, or exit the macro subsystem by entering the EXIT command again.

HELP command

The HELP command allows you to obtain a quick syntax reference for any or all commands valid in the macro subsystem. If syntax help is sought for only one command, the information is shown on the CRT display. If syntax help is sought for all commands in the macro subsystem, the information is printed on the computer's built-in printer.

For example:

```
HELP DEFINE ;

DEFine name [<praml ["dflt1"] ...>] "body"
```

The command name is displayed showing the first three letters as capital letters with the remainder of the command name shown in lower case. This indicates that only the first three letters of the command need be entered.

Items enclosed in square brackets [ ] represent optional parameters that may be entered with the command. Items represented in lower case within the brackets are short notes indicating the type of data or parameter expected. All other characters (such as ` ` and >) should be entered exactly as shown.

These help messages are intended to be a quick reference for the experienced user. The help message for each command is listed with the syntax diagrams in the syntax reference chapter of this manual. In this way you can quickly learn to associate the short parameter descriptions with the longer explanations provided with the syntax diagrams.

LIST command

The list command allows you to list the macros currently defined. Entering the LIST command causes each macro definition to be printed on the current system printer. Additionally, you may obtain a listing of the contents of the current process file, or a listing describing any device contained in computer memory.

SAVE command

The SAVE command allows you to store the current macro command definitions in an EWE file called a macro file.

Process Subsystem

Overview

The process subsystem is the portion of the Graphics Editor where the layers that compose the drawing area of the display are defined and the user units of the drawing area are set. This information may be stored in an EWE file called a process file, and may be recalled (loaded into the system) at any time to create other drawings using the same layer definitions and units.

Many different process files may be created and stored, but only one may be loaded into the system at a time, thus defining the layers and units. The process subsystem is entered through the Graphics Editor by entering the PROCESS command. This also allows you to load a specific process file.

Some basic concepts are provided in this section to help you understand how the subsystem works. The commands available in the process subsystem are then described, and an interactive example is provided to let you gain experience with the concepts and commands.

Concepts

Layers

In the edit subsystem concepts, you found that a drawing is created by adding components to one of 255 clear sheets, called layers. The sheets are arranged in a stack such that you can view the components on any layer or on any combinations of layers. However, before a layer can be drawn on it must be defined in the process subsystem.

Defining the layer involves specifying:

- a layer number
- a layer label
- the line type (with which components drawn in this layer are displayed, such as solid, broken, dashed, etc.)
- the layer display type
- the color with which components in the layer are displayed (color systems only)
- the plotter pen number with which components in the layer are plotted
- the font size with which inames and tnames of the components are displayed
- iname and tname display option (causing inames and/or tnames of all components drawn in the layer to be plotted on the display at all times)

While up to 255 layers can be defined at one time, you need define only the layers you intend to use. The EGS-45 software requires that layer 0 is always defined, and that it is reserved by the system for displaying instance components. Also, HP reserves layers 1-100 for factory defined process files.

Units

In the edit subsystem concepts, you found that x,y locations can only be entered on user grid points. You also found that the foundation of this grid is the system grid. In the process subsystem, you assign a physical distance (a unit, such as 1 inch) to the distance between system grid points, and thus determine the units of the drawing.

For example, by specifying that the distance between 100 system grid points equals 1 inch, you have defined the size of the total drawing area to be from −163.83 to 163.83 inches in both the X and Y directions (since the the system grid ranges from −16383 system units to 16383 system units in each the x and the y directions). Inches then becomes the user units, and locations may be accessed to a resolution of .01 inch.

Layer Display Types

One of three different display types must be specified when defining a layer: detail type, interconnect type, and symbolic type. The WINDOW command in the edit subsystem determines the display mode in the drawing area. When the detail display mode is set, components entered in layers defined as detail layers are displayed, along with components entered on layers defined as interconnect layers. When the symbolic display mode is set, components entered on layers defined as symbolic layers are displayed, along with components entered on layers defined as interconnect layers. When the interconnect mode is set, only components entered on layers defined as interconnect layers are displayed.

The layer display modes thus allow you to have two separate drawings in the drawing area, one on layers defined as symbolic layers, and one on layers defined as detail layers. Components entered on interconnect layers would be displayed in both drawings; hence the name interconnect. Using the WINDOW command, you can specify which drawing is to be viewed.

Process Subsystem Commands

EQUATE command

The EQUATE command is used to define each layer that is to be used to create a drawing. With the EQUATE command you specify:

- the number of the layer that is being defined
- an 8 character label for the layer
- the layer display mode
- the line type used to display components entered on the layer or the color with which to display components entered on the layer (systems with color CRTs only)
- the plotter pen number that is used to plot components entered on the layer
- that the inames and/or tnames assigned to components in that layer are plotted in the display area at all times
- the size of the iname/tname characters that are plotted in the display area

Only the characters A-Z, 0-9, (_), (*), (∧), ($), ([), (]), (.), (,), (+), (−), (<), (>), (/) and (\) can be displayed with the iname/tname display option, even though a larger number of characters can be assigned and displayed with the IDENTIFY and TRACE commands.

A set of layer definitions and user units definitions are provided for you in the process file, PROCDATA, that is shipped with your system. You may change the layer and units definitions contained in this process file to meet the needs of your applications or you can create your own process file or you can create a new process file from scratch. The layer definitions and the user units definitions contained in PROCDATA are listed below. Use this as an example to base your own layer definitions.

Resolution: 1 UM is equal to 1 system grid point.

Process file is EWSYS:PROCDATA

LAYER #	PEN #	LABEL	COLOR	LINE TYPE	LAYER TYPE	IDENTIFY NAME disp	IDENTIFY NAME size	TRACE NAME disp	TRACE NAME size
0	0	INSTANCE	White	Dotted	Boundary	OFF	1	OFF	1
1	1		Red	Solid	Detail	OFF	1	OFF	1
2	2		Yellow	Solid	Detail	OFF	1	OFF	1
3	3		Green	Solid	Detail	OFF	1	OFF	1
4	4		Aqua	Solid	Detail	OFF	1	OFF	1
5	5		Blue	Solid	Detail	OFF	1	OFF	1
6	6		Purple	Solid	Detail	OFF	1	OFF	1
7	1		Red	Dotted	Detail	OFF	1	OFF	1
8	2		Yellow	Dotted	Detail	OFF	1	OFF	1
9	3		Green	Dotted	Detail	OFF	1	OFF	1
10	4		Aqua	Dotted	Detail	OFF	1	OFF	1
11	5		Blue	Dotted	Detail	OFF	1	OFF	1
12	6		Purple	Dotted	Detail	OFF	1	OFF	1
13	1		Red	Broken	Detail	OFF	1	OFF	1
14	2		Yellow	Broken	Detail	OFF	1	OFF	1
15	3		Green	Broken	Detail	OFF	1	OFF	1
16	4		Aqua	Broken	Detail	OFF	1	OFF	1
17	5		Blue	Broken	Detail	OFF	1	OFF	1
18	6		Purple	Broken	Detail	OFF	1	OFF	1
19	1		Red	Dashed	Detail	OFF	1	OFF	1
20	2		Yellow	Dashed	Detail	OFF	1	OFF	1
21	3		Green	Dashed	Detail	OFF	1	OFF	1
22	4		Aqua	Dashed	Detail	OFF	1	OFF	1
23	5		Blue	Dashed	Detail	OFF	1	OFF	1
24	6		Purple	Dashed	Detail	OFF	1	OFF	1
30	1		Red	Solid	Detail	ON	10	ON	10
31	2		Yellow	Solid	Detail	ON	10	ON	10
32	3		Green	Solid	Detail	ON	10	ON	10
33	4		Aqua	Solid	Detail	ON	10	ON	10
34	5		Blue	Solid	Detail	ON	10	ON	10
35	6		Purple	Solid	Detail	ON	10	ON	10
40	1	INTER1	Red	Solid	Interconnect	OFF	1	OFF	1
41	2	INTER2	Yellow	Solid	Interconnect	OFF	1	OFF	1
42	3	INTER3	Green	Solid	Interconnect	OFF	1	OFF	1
43	4	INTER4	Aqua	Solid	Interconnect	OFF	1	OFF	1
44	5	INTER5	Blue	Solid	Interconnect	OFF	1	OFF	1
45	6	INTER6	Purple	Solid	Interconnect	OFF	1	OFF	1
50	1	SYMB1	Red	Solid	Symbolic	OFF	1	OFF	1
51	2	SYMB2	Yellow	Solid	Symbolic	OFF	1	OFF	1
52	3	SYMB3	Green	Solid	Symbolic	OFF	1	OFF	1
53	4	SYMB4	Aqua	Solid	Symbolic	OFF	1	OFF	1
54	5	SYMB5	Blue	Solid	Symbolic	OFF	1	OFF	1
55	6	SYMB6	Purple	Solid	Symbolic	OFF	1	OFF	1

Unused memory = 270 Kbytes

EXIT command

The EXIT command allows you to leave the process subsystem and return to the Graphics Editor. If the definition of any layer has changed or if the user units definition has changed since the process subsystem was entered, and these changes have not been saved in a process file, the system issues a warning message. You can either save the process file with the SAVE command, continue modifying the layer definitions or the units, or enter the EXIT command again to exit the process subsystem without saving the file. The layer definitions and user units defined in the process subsystem are the active layer and unit definitions, regardless of whether the process file has been saved or not.

HELP command

The HELP command allows you to obtain a quick syntax reference for any or all commands valid in the process subsystem. If syntax help is sought for only one command, the information is shown at the top of the display. If syntax help is sought for all commands in the edit subsystem, the information is printed on the current system printer.

To explain the syntax representation that is provided, consider the response to entering:

```
HELP EQUATE

EQUate [:<S|D|I>] [:Ccolor] [:Ppen] [:M<S|D|B|L>] [:T<I|T>[font]]
[label] layer
```

These help messages are intended to be a quick reference for the experienced user. The help message for each command is listed with the syntax diagrams in the syntax reference chapter of this manual. In this way you can quickly learn to associate the short parameter descriptions with the longer explanations provided with the syntax diagrams.

LIST command

The LIST command allows you to list the contents of the process file currently loaded into the system. The process file name and the volume on which it is stored are also provided. Additionally, you may list the contents of the current macro file and obtain a listing describing any devices resident in computer memory.

The listing obtained is printed on the current system printer.

SAVE command

The SAVE command allows you to store the layer definitions and the user units defined in the process subsystem in an EWE file, called a process file.

UNITS command

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 from -16383 to $+16383$ in both the x and y directions, you set the maximum drawing size and the resolution at which locations can be accessed. For example, setting 10 system grid points equal to 1 foot, sets the size of the total drawing area to span from -1638.3 feet to $+1638.3$ feet in both the X and Y directions. Since there are 10 system grid points per foot, you can access locations to a resolution of 0.1 foot (only x,y locations corresponding to a user grid point may be specified).

If the user units of the process file do not match the user units of a device in computer memory, the system will issue a warning. You may either change the user units definition to match those of the device or you can ignore the warning and allow the system to convert the user units of the device into the current user units of the process subsystem.

Table of Contents

Chapter 3
The File Utilities

Introduction

In Chapter 2, the concepts of files, volumes, and environments were introduced. You learned that EGS-45 utilizes many special types of files (EWE files such as device files and macro files) that may only be accessed by the EGS-45 software.

The File Utilities module allows you to access these special files and volumes while providing the capabilities to rename files, move files from volume to volume, and make backup copies of files and volumes.

The File Utilities also provides the capabilities to remove individual files from a volume. This is useful for removing unneeded files from a volume, thus conserving mass storage space.

This chapter discusses the use of the File Utilities module. Several basic concepts are discussed, such as files and file specifiers (identifiers). Next the capabilities of the different functions provided by the module are discussed while several examples are provided to show how these capabilities are used.

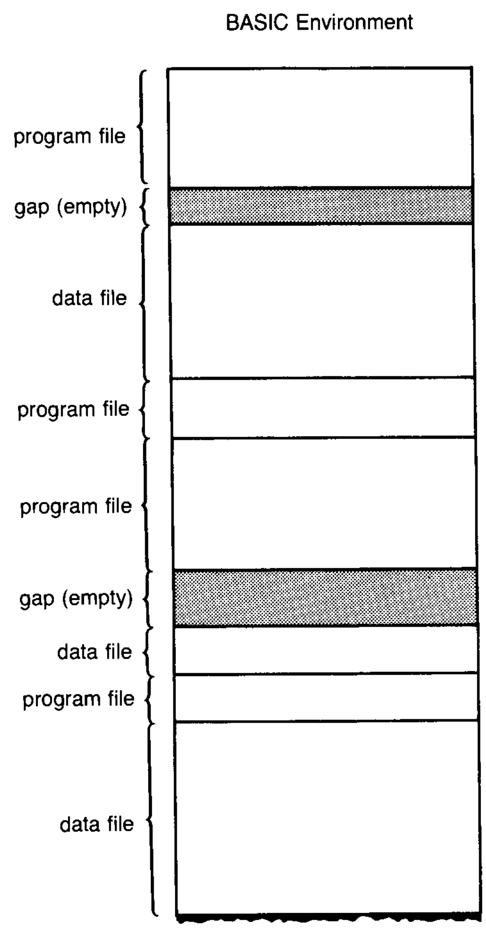
The foldout systems software diagram supplied at the beginning of Chapter 1 shows the different capabilities offered by this module.

Concepts

Files, Volumes, and Environments

A file is a discrete "chunk" of associated information that is stored on disc or tape. This "chunk" of information may be a computer program or data required by a program. In the EGS-45 system files are separated into two groups, called environments: the BASIC Environment and the Editor Workstation Environment (EWE)

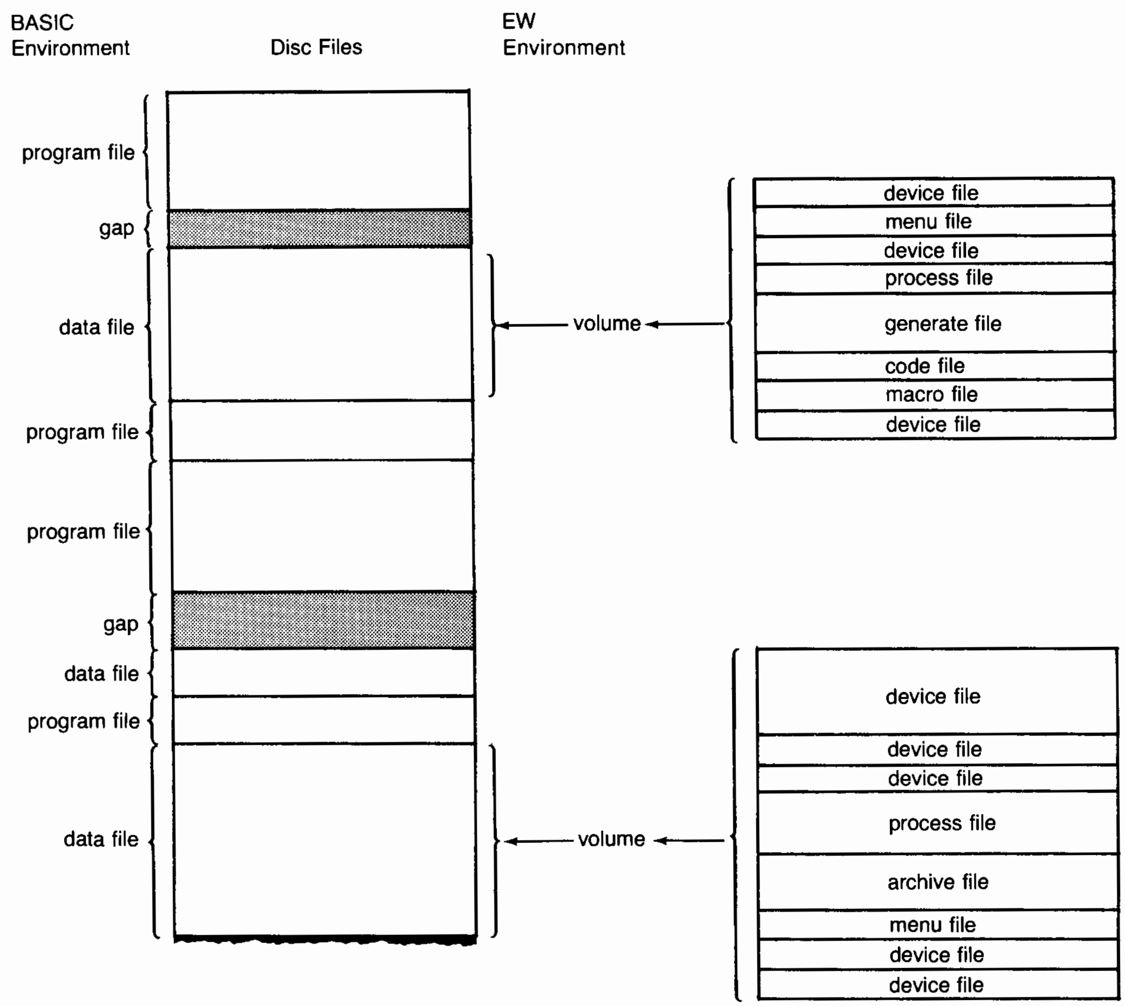
Physically, the computer stores BASIC environment files in a rather complex manner. Conceptually, BASIC program and data files may be thought of in terms of the drawing below. Here a file is shown as a continuous block of information. Files may be contiguous or gaps may exist between files on the disc. The size of the file is measured by the amount of area on the mass storage media that it occupies.



Files existing in the BASIC environment are accessed with BASIC language statements and commands. These files contain the EGS-45 programs and some of the data required by these programs. You may obtâin a list of the BASIC files contained on a particular mass storage medium with the BASIC language CAT statement (see your computer's operating and programming manuals). Additionally, you may obtain a list of the BASIC files on a mass storage medium via the capabilities of the File Utilities module.

In contrast, the files created and used by the EGS-45 software are said to be in the Editor Workstation environment (abbreviated EWE). Examples of EWE files are device files, process files, archive files, and generate files. Files existing in the EW environment may be accessed with the capabilities of the File Utilities module. Additionally, programming tools are provided with EGS-45 which allow you to access EWE files with your own BASIC language application programs (see Chapter 6 for information).

EWE files are stored in an EWE volume. An EWE volume is simply a BASIC environment data file. The EGS-45 software divides the volume into a series of individual cubbyholes, which hold the individual EWE files. The size of the cubbyhole is dependent on the size of the EWE file which is stored there. The following drawing shows how the files stored on a single disc are viewed in both the BASIC and EW environments.



Volumes may be used to separate information into logical groups. For example, a user can keep his personalized process, macro, and menu files as well as his drawings on his personal volume. If a user accesses only one volume (his volume) he is less likely to delete or destroy the files of another user.

Volumes are assigned a name and a volume number either of which may be used to identify the volume. The system volume number is #3 and its volume name is EWSYS.

The system also considers the CRT, the keyboard, and the system printer to be volumes. The following volume names and volume numbers are reserved by the system:

Volume Number	Volume Name	Description
#1	CONSOLE	Specifies the system CRT
#2	SYSTERM	Specifies the system keyboard
#3	EWSYS	Specifies the system volume
#6	PRINTER	Specifies the system printer

When creating a drawing in the edit subsystem, it is possible that no volume has enough room to store your drawing. If the drawing cannot be stored, all modifications to the drawing are lost when exiting the Graphics Editor. Therefore, you must make sure that you have enough room left in a volume to store your work.

To avoid losing data because of a full volume, it is recommended that you create a small (about 1000 records) "emergency" volume. Use this volume only when you are attempting to save a drawing and all other volumes are full. This volume serves as an escape valve to prevent data loss. Immediately after using your emergency volume, exit the Graphics Editor. Use the File Utilities module to create a new volume or to bring an existing volume with space on line. Then transfer your drawing from the emergency volume into the new volume. Next, remove your drawing from the emergency volume. If these suggested procedures are followed, you will again have space to store a drawing when all other available volumes are full.

Suffixes

As you have already learned, there are many types of EWE files used by the system (such as device files and archive files). To allow the system to differentiate between file types, a 4 character suffix is automatically appended to the file name when the file is created by the Graphics Editor or its subsystems.

These suffixes allow you to have two different types of files with the same name. For example, when an archive file is created from a device file, the archive file is assigned the same name as the device file. If a suffix was not assigned to each file name and if both files were stored on the same volume, the system could not tell which was the device file and which was the archive file. In the Graphics Editor, it is not necessary to use these suffixes with the file name since the system knows which type of file you are referencing from the command accessing the file. For example, if EDIT EXAMP; is entered, the system knows that EXAMP is the name of a device file, since only a device file may be edited.

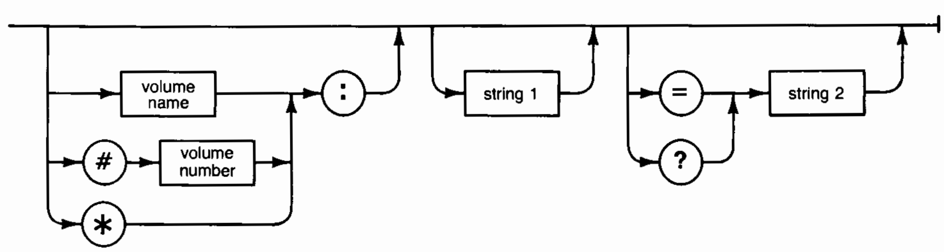
You will use the file suffixes in the File Utilities modules and possibly in the optional application modules. The suffixes used by the system and the file types that they represent are listed below.

Suffix	file type
.ARC	archive file
.DEV	device file
.GEN	generate file
.MAC	macro file
.MEN	menu file
.PRO	process file
.PLOT	plot file
.CODE .OS .STARTUP .FILER	system program code files. Do not delete or remove these files.

Care must be taken to preserve the suffix of the file when using the File Utilities. If the suffix is changed or removed the Graphics Editor and its subsystems cannot recognize the file type and will not be able to use the file. For example, changing the device file DRAWING.DEV to ARTWRK.DEV preserves the file type. If the suffix .DEV had been omitted, the file would remain on your mass storage medium but could not be accessed by the Graphics Editor.

File Specifiers and Wildcards

When working within the File Utilities Module, you are often asked to identify a file by entering a file specifier. A file specifier is a string of characters which consists of a volume specifier, a file name, the file type suffix, and several optional parameters that identify like groups of files within a volume. A volume specifier identifies a particular volume. It consists of the volume name or the volume number, followed by a colon (:). The syntax for the file specifier is shown below.



Item	Description/Default
volume name	Name of the volume. The volume name may be the name of either an EWE volume which contains EWE files or an EWE volume consisting of hardware (i.e. PRINTER, CONSOLE, etc).
volume number	Number of the volume. The volume number may be the number of either an EWE volume which contains EWE files or an EWE volume consisting of hardware (i.e., #1, #2, etc).
*	Specifies the system volume, EWSYS.
string 1	Set of characters that identifies one or several EW files. This string of characters is the file name plus the file type suffix if no wildcards are specified. This string may be empty (i.e., no characters entered) or may be a portion of a file name when a wildcard is specified.
= or ?	These are wildcard characters that allow you to specify many EW files at once. Wildcards are used only in the Filer section of the File Utilities module and are discussed in the sections below.
string 2	Set of characters used with wildcards to help identify like files. This string may also be empty if a wildcard is specified.

The wildcard characters, (=) and (?), are used to specify subsets of EW files within a volume. Used only in the Filer portion of the File Utilities, wildcards cause the function being performed to act on all files meeting the file specifications. For example, the file specifier EWMINE:DOC=.DEV specifies to act on all files on the volume EWMINE that begin with the string DOC and end with the characters, DEV. The file specifier EWMINE:DOC?.DEV specifies to act on the same files, the only difference being that a prompt is displayed that requests string verification before affecting each file meeting the specified criteria.

Examples showing the use of file specifiers and wildcards are provided below. In each case, the prompt for the example is the prompt for the Remove File branch of the Filer (part of the File Utilities). Use the sample file directory for the fictional volume EW1:

```
EW1:
BIGDRAW.DEV      6  23-Jun-81   6   512   Devicefile
MOLD.DEV         8  29-Jun-81  12   512   Devicefile
USELESS.CODE    15  10-Jul-80  20   512   Codefile
LARGE.DEV        5   3-Oct-78  35   512   Devicefile
NEVERMOR.MAC     3   8-Oct-81  38   512   Macrofile
NEEDLESS.PRO     3  15-Oct-81  41   512   Processfile
```

Example
System prompt: `Remove what file?`

Your response: `EW1:MOLD.DEV`

This causes the file MOLD.DEV to be removed from the volume EW1, if it exists.

Example
System prompt: `Remove what file?`

Your response: `EW1:NE?`

This causes the system to find all files on the volume EW1 whose name begins with NE (NEVERMOR.MAC and NEEDLESS.PRO). The system then prompts you with each file name to see if you wish to remove the file. Responding Y (for yes) to the prompt causes the file to be removed. Responding N (for no) to the prompt causes the file to not be removed.

Example
System prompt: `Remove what file?`

Your response: `EW1:=`

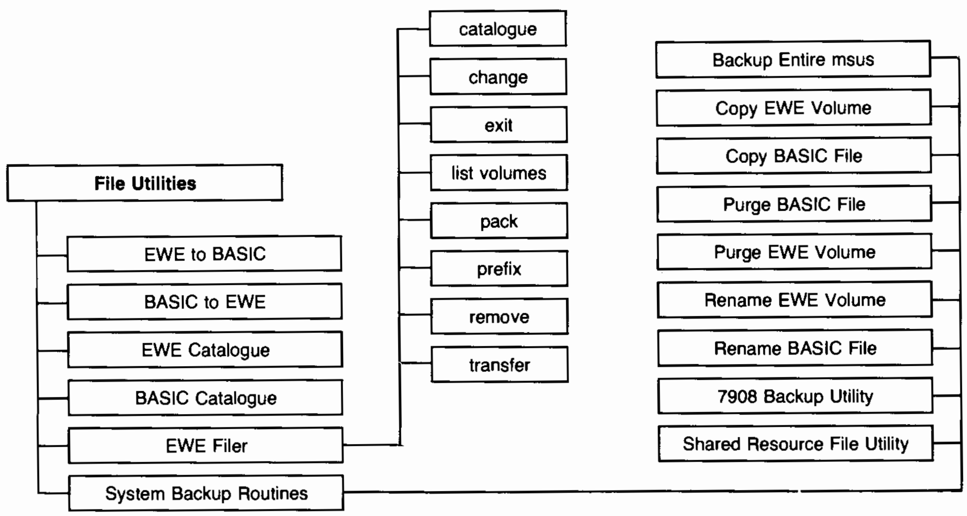
This causes the system to remove **every file** on the volume EW1 since a wildcard (=) was specified without any limitations (no strings are entered with the wildcard). Using either wildcard within a file specifier tells the system to operate on every file in the volume and limit the selection of files to those that contain the specified strings. If no limiting strings are specified the system performs the specified function on every file in the volume. **Caution should be exercised when using wildcards with file specifiers.**

Msus

Several of the File Utilities options require you to enter a mass storage unit specifier (msus) which identifies a specific mass storage device. The msus consists of a colon, a single character specifying the type of device, and the select code and address of the device. A list and full explanation of msus is provided in your computer's Mass Storage manual.

The File Utilities

The File Utilities module connects to a tree structured menu. Each "branch" of this tree provides specific capabilities and is discussed in the sections below.



To enter information requested by the File Utilities, type the required information on the computer keyboard and press CONTINUE. Until CONTINUE is pressed, you may alter the information you type by using the left-right arrow keys and the insert-delete character keys. This procedure is slightly different for the Filer branch of the File Utilities. In the Filer branch, using the back space key or the left arrow key causes the characters to be deleted as you "back over" them.

Return to Manager

Selecting Return to Manager allows you to exit the File Utilities module. This allows you to select any of the other EGS-45 modules.

EWE to BASIC

Selecting EWE to BASIC allows you to transfer an EW file into the BASIC environment, thus allowing it to be accessed by a BASIC program. This is achieved by creating a BASIC data file and transferring the contents of the EWE file into the BASIC file. Once this branch has been selected the system prompts you for the file specifier identifying the file to be transfered. Enter the file name and press CONTINUE.

If no response is entered the system assumes that no transfer is desired and exits you from the File Utilities module.

If the EWE file specifier that you enter does not exist, the system assumes you made a mistake entering the file name. It again asks you to enter the file specifier identifying the file to be transferred.

Next, the system prompts you for the name of the BASIC file in which to transfer the EWE file. The BASIC file is created on the mass storage device containing the system software. To specify a diffent mass storage device, simply append the msus of the device to the BASIC file name. For example, MINE:F8 specifies to create a BASIC file named MINE on an HP 9885 disc located at select code number 8.

- If the file already exists in the BASIC environment, you are asked if you want to replace the old file with the file being transferred. Responding N, for no, causes the system to prompt you for the BASIC file name again.
- If no name is input, the system assumes that you do not wish to continue the transfer and prompts you for the file specifier identifying the EWE file to be transferred.
- If the file name PRINTER: is input, no BASIC file is created. Instead, a listing of the EWE file is printed on the system printer.

BASIC to EWE

Selecting BASIC to EWE allows you to transfer a BASIC environment file into the EW environment, thus allowing it to be accessed by EGS_45. This is achieved by creating an EWE file and transferring the contents of the BASIC file into the EWE file. Once selected, the system prompts you to enter a file specifier identifying the EWE file and volume in which to transfer the BASIC file.

- If no file name is input and only (CONTINUE) is pressed, the system assumes that no transfer is desired and exits the File Utilities module.
- If the name PRINTER: is input, no EWE file is created. A listing of the BASIC file specified is printed on the system printer.
- If the file already exists in the EW environment, you are asked if you want to replace the old file with the file being transferred. Answering no, causes the system to again prompt you for the file specifier identifying the EWE file in which to transfer.

Next, the system prompts you for the name of the BASIC file to be transferred. If you do not input a file name the system returns to the first prompt and again asks you for the file specifier identifying the EWE file in which to transer the BASIC file. If the BASIC file name that you enter does not exist, the system assumes you made a mistake entering the file name, and asks you for the BASIC file name again.

EWE Catalogue

The EWE Catalogue branch allows you to get a listing of all EWE files contained in an EWE volume. Selecting this branch causes the system to display a menu of all EWE volumes present on the system. Selecting an individual volume causes the system to print a listing of all EWE files in that volume on the current system printer. To return to the File Utilities menu, select Return to Manager.

BASIC Catalogue

The BASIC Catalogue branch allows you to get a listing of all BASIC files located on a particular mass storage device. Selecting this branch causes the system to display a menu of all mass storage devices present on the system. Selecting an individual device causes the system to print a listing, on the system printer, of all BASIC files on that mass storage device. To return to the File Utilities menu, select Return to Manager.

EWE Filer

The Filer branch allows you to perform normal file management functions on the EWE files within an EWE volume. It allows you to transfer EWE files between volumes, remove unwanted EWE files, change an EWE file name, list out all of the EWE files on an EWE volume, list all EWE volumes on the system, and compact all of the files on an EWE volume so that you can conserve space within the volume.

When working in the Filer, both upper case and lower case letters may be entered. The system automatically converts all lower case characters entered into upper case characters. Therefore, you may enter whichever case character that is convenient.

Selecting EWE Filer causes the system to display a menu of its functions on the CRT. Each of these functions and its capabilities is discussed in the sections below.

Exit

Selecting this function allows you to exit the EWE Filer and causes the File Utilities menu to be redisplayed.

List Volumes

Selecting this function causes the system to display a list of all volumes currently recognized by the system. This listing shows the volume number and the volume name. The asterisk (*) identifies the system volume, EWSYS while the pound sign (#) identifies all other EWE volumes containing EWE files. Hardware volumes (such as PRINTER:) are not identified with either character. The system volume is the default volume for accesses to files. You can change the default volume (for use in the Filer only) with the Prefix function in the Filer.

A typical display resulting from selection of the Volumes function is:

```
Volumes on-line:

1    CONSOLE:
2    SYSTERM:
3  * EWSYS:
4  # EW1:
6    PRINTER:
```

To obtain a printed copy of this listing, simultaneously press:

CONTROL ko

Catalogue

The Catalogue function allows you to list all files or a subset of all files on a specified volume. The catalogue listing may be directed to the system printer, or to an EW file in a specified volume. The default destination of the listing is the system CRT.

A sample catalogue listing is shown below. It consists of:

- the name of a volume specifier identifying the volume being listed
- a list of all file names contained in the volume
- each file's size in blocks (one block equals two records; one record equals 256 bytes)
- the last modification date,
- the starting block address of the file within the volume,
- the number of bytes in the last block of the file,
- the file type (assigned by the system and determined by the file name suffix).

```
File name            File size    Last update       Addr       LBS       File type
EWSYS:
SYSTEM.OS                  40      13-Nov-81          26       512        Codefile
SYSTEM.FILER ·             30      12-Nov-81          66       512        Codefile
STREAM.STARTUP              1       3-Dec-81          96         1        Datafile
MENUDATA.MEN                2       9-Sep-81          97       468        Datafile
CEDIT.CODE                 11      20-Nov-81          99       512        Codefile
MACRDATA.MAC                1      30-Sep-81         110       246        Datafile
PROCDATA.PRO                3      12-Nov-81         111       120        Datafile
GEDIT.CODE                267      12-Nov-81         114       512        Codefile
EXAMP.DEV                   1      23-Nov-81         381       234        Datafile
< UNUSED >                  2                        382
CIRC.DEV                    2      23-Nov-81         384        46        Datafile
LINES.DEV                   1      23-Nov-81         386       360        Datafile
POLS.DEV                    1      23-Nov-81         387       252        Datafile
GRIDS.DEV                   3      23-Nov-81         388       236        Datafile
EXAMP2.DEV                  1      23-Nov-81         391       234        Datafile
EXAMP3.DEV                  1      23-Nov-81         392       234        Datafile
ARKS.DEV                    1       1-Dec-81         393       252        Datafile
NOTENEW.DEV                 1       1-Dec-81         394       216        Datafile
TEXTNEW.DEV                 1       1-Dec-81         395       216        Datafile
RECSNEW.DEV                 1       1-Dec-81         396       252        Datafile
< UNUSED >                  3                        397
19/19 files<listed/in-dir>, 369 blocks used, 5 unused, 3 in largest area
```

The first item listed in the catalogue is the volume specifier identifying the volume being catalogued (EWSYS:). The first column contains the complete file name of the files stored on the volume (such as EXAMP.DEV and GEDIT.CODE). The second column contains a numeric value specifying the size of the file, in blocks (1 block = 512 bytes = 2 records). The third column identifies the date when the file was last modified or stored. The fourth column identifies the block number on the volume where the file begins (for example, EXAMP.DEV is a one block long file beginning at block number 381 on the system volume, EWSYS). The fifth column specifies the number of byte used in the last block of the file (this specifies the number of bytes leftover in the 512 byte block). The last column identifies the type of file that is stored. Note that device, macro, process, and menu files are listed as Datafiles.

The last entry in the file name column is, <UNUSED>, which specifies that the area on the rest of the media has not been used. The number of blocks left on the media is listed along with the beginning block number of the unused area.

<UNUSED> also appears in the middle of the list of files stored on EWSYS. This indicates that a gap is present between the files. This is created when an existing file has been removed from the volume. Many such gaps may exist on a volume. By using the Pack function of the Filer you may pack all files on a volume together and eliminate this wasted space.

The information on file size and location should help you determine the size of the volumes and files you create in other sections of the File Utilities.

Selecting the Catalogue function causes the system to prompt:

```
Catalogue of what volume?
```

Respond by entering a file specifier identifying the volume (and optional subset of files) that you want catalogued. The file specifier must contain the volume name or volume number followed by a colon. Using a wildcard and the associated limiting character strings, you may obtain a subset of the full directory listing of a volume.

For example, responding

```
EWSYS:N=.DEV
```

causes a catalogue listing of all files on the volume EWSYS, whose name begins with N and ends with .DEV. If you wish to direct the listing to any other device or file other than the system CRT, you may specify a destination volume and file with the file specifier. Source volume and file information is separated from destination file and volume information by a comma (,).

To obtain a printed copy of a listing displayed on the system CRT, simultaneously press:

CONTROL k0

Several examples are provided below to help you understand the use of the Catalogue function.

Example:

```
Catalogue of what volume?

EWSYS:
```

A listing of every file in the volume EWSYS is displayed on the system CRT.

Example:

```
Catalogue of what volume?

#4:FI=EV
```

All files in volume number 4 whose file names begin with the characters FI and end with the characters EV are displayed.

Example:

```
Catalogue of what volume?

*:FI=EV,PRINTER:
```

A listing of all files in the system volume whose file names begin with the characters FI and end with the characters EV is printed on the current system printer.

Example:

```
Catalogue of what volume?

#4:,#5:TRASH
```

A listing of all files on volume number 4 is written into an EWE file named TRASH on volume number 5.

Change

The Change function allows you to change the name of one or more files in a specified volume. The function requires that two file specifiers be entered: a source file specifier that identifies all files whose names are to be changed and a destination file specifier that identifies the changes to be made to the source file names. A comma separates source file information and destination file information. Wildcard specifiers are allowed. However, if wildcards are used, they must be used in both the source file specifier and the destination file specifier.

Selecting Change causes the system to display the prompt:

```
Change what file?
```

Specifying only a source file causes the system to prompt you for a destination file specifier. Specifying both the source and the destination file specifiers causes the name of the source file to be changed to the name of the destination file.

Several examples are provided below to help you understand the use of the Change function.

Example:

```
Change what file?

#5:TEST.DEV,#5:PASS.DEV
```

The name of the device file TEST, DEV on volume number 5 is changed to PASS, DEV.

Example

```
Change what file?

EWSYS:NO=V,EWSYS:YES=Z
```

This response causes the system to search for all files on the volume EWSYS whose names begin with the characters NO and end with the character V. It then replaces the first two characters (NO) of those file names with the characters YES. It also replaces the trailing character V with the character Z. For example, if EWSYS originally contained the file NONO.DEV, its new name would be YESNO.DEZ.

Example:

```
Change what file?

=,T=N
```

The system searches the default volume for all files, since no limiting search characters are present with the wildcard in the source file specifier. It then changes each file name such that the character T is added to the beginning of each file name and the character N is added to the end of each file name. The destination file specifier indicates to replace the characters preceding the wildcard in the source specifier (none, in this case) with the characters preceding the wildcard in the destination specifier (T, in this case). This same logic is applied to the characters following the wildcard in the source and destination file specifiers.

For example, if the default volume contained the file name, BIG.DEV, the new file name would be, TBIG.DEVN.

Remove

The Remove function allows you to delete any EW file from a specified volume. The system requires that you enter the file specifier identifying which file is to be removed from the volume. Using wildcards allows you to remove several files with one entry. When a wildcard is used with the Remove function, the system displays a list of all files identified. It then prompts you to see if you want to remove all of the files identified. This prompt appears as:

```
Update directory (y/n)?
```

Answering Y, for yes, causes the system to remove all files specified with the wildcard. Answering N, for no, causes the system to exit the Remove function without removing any files.

The examples provided below help explain how the Remove function works.

Example:

```
Remove what file?

FILE.DEV
```

The file FILE.DEV is removed from the default volume.

Example:

```
Remove what file?

A=DEV
```

A list of all files on the default volume whose names begin with A and end with DEV is displayed. The system next prompts you to find out if you wish to update the directory. Answering Y, for yes, causes the system to remove the listed files. Answering N, for no, causes the system to exit the Change function without removing any files.

Example:

```
Remove what file?

EW1:=
```

A list of all files in the volume EW1 is displayed. If you specify to update the directory, every file in the volume EW1 is removed.

Transfer

The Transfer function allows you copy the contents of the specified source file to the specified destination. This may be another EWE file or the system printer. The system requires that you enter a source file specifier and a destination file specifier. Wildcards are allowed, thus allowing you to copy several files with one entry.

Selecting the Transfer function causes the system to display the prompt:

```
Transfer what file?
```

Entering the source file specifier and the destination file specifier (separated by a comma) causes the system to transfer the file(s) specified.

Entering only a source file specifier causes the system to display the prompt:

```
To where?
```

You should then enter the destination file specifier. If a file already exists with the file specifier that you enter, the system displays a message to that effect and asks you if you wish to remove the existing file. Responding Y, for yes, causes the system to remove the existing file and replace it with the file specified with the transfer function. Responding N, for no, causes the system to exit the Transfer function.

It is often convenient to transfer a file without changing the name and without re-typing the file name. For example, you might want to transfer several device files from your volume to the volume of a fellow worker. The Transfer function enables you to do this by allowing the character $ to replace the filename, or portion of the file name in the destination file specification. Several examples are provided below to help you understand the use and capabilities of the Transfer function.

Example:

```
Transfer what file?

EW1:MY.DEV

To where?

EWSYS:YOUR.DEV
```

The contents of the file MY.DEV on the volume EW1 is copied into the file YOUR.DEV that the system creates on the volume, EWSYS.

Example:

```
Transfer what file?

PROC.PRO,EW1:PROC.PRO
```

The contents of the file PROC.PRO on the default volume is copied into the file PROC.PRO on the volume, EW1. Responding to the prompt with:

```
PROC.PRO,EW1:$
```

produces the same result. Specifying $ instead of the destination file name causes the system to use the source file name as the destination file name.

Example:

```
Transfer what file?

EWSYS:BIG=.MAC,EW1:LACK=.MAC
```

This response causes the system to find each file in the volume EWSYS whose name begins with the characters BIG and ends with the characters .MAC, and copy it to the volume EW1. It replaces the original file name characters BIG with the characters LACK in the destination file names.

Example:

```
Transfer what file?

*:MY?.DEV, EW1:$
```

This response causes the system to find each file in the system volume whose name begins with the characters MY and end with the characters as each file is located to see if you desire that particular file to be transferred. Responding yes causes the system to copy the file from the system volume to the volume, EW1.

Because $ is specified as the destination file name, the file name assigned to the copied file on volume EW1 is the same name as the original file.

Example:

```
Transfer what file?

EW1:=, EW2:$
```

Every file on the volume EW1 is copied to the volume EW2. Because $ is specified as the destination file name, the names of the files copied to EW2 are identical to the names of the files on EW1.

Prefix

The Prefix function allows you to change the default volume used by the EWE Filer. This is valid only in the EWE Filer. Whenever a volume is not explicitly or implicitly stated, the system uses the default volume as the volume to be acted upon. The default volume is the system volume, EWSYS, when the Filer is selected.

Selecting Prefix causes the system to prompt you for the volume specifier identifying the new default volume. Responding with the volume name or volume number followed by a colon (:) causes the system to assign that volume as the default volume. Responding with only a colon (:), causes the system to display the current default volume.

Pack

The Pack function allows you to pack the files contained in the specified volume such that gaps left between existing files are minimized. This allows you to utilize all of the available space in a given volume.

Selecting Pack causes the system to prompt you for the volume specifier of the volume to be packed. The volume specified must be present and recognized by the system. The system then prompts you:

```
Are you sure you want to pack (volume specifier) (y/n)?
```

Answering N, for no, causes the system to exit the Pack function. Answering Y, for yes, causes the volume to be packed. As each file is moved, its name is displayed on the CRT. The system displays a message telling you when the pack is complete.

System Backup Routines

The purpose of this section is to describe how you may use the capabilities of the File Utilities module to maintain the software and data files that compose EGS-45. Generally speaking, the term backup means making a copy of files and/or data stored on mass storage media. Individual files, individual volumes, or everything on a disc may be copied to provide a backup.

Drawings, library parts, and software stored on mass storage media represent a large investment of time and money. And although the quality and reliability of the mass storage media is very high, unforeseen circumstances can cause the destruction of the mass storage media (such as fire, spilled coffee, smoke contamination, dust, etc). For this reason it is important that your investment in software and time is not lost. To minimize the possible loss, it is recommended that periodic backups (copies) of the entire system and/or individual files be made.

The frequency and the extent of these backups is different for each user of the system. It is largely determined by answering the question, "How much can I afford to lose?" Once the data on a mass storage media is lost, it can never be recovered.

To keep from having to repurchase the original software package, a backup of the software should be made and kept in a location separate from the active system software and volumes.

Additionally, as more and more time is invested in creating drawings and library parts, backups of the volumes containing these devices should be made.

The system provides the means for creating backups through the System Backup Routines branch of the File Utilities module. You may backup an entire mass storage media, an individual volume, or an individual BASIC file. This branch also allows you to purge (remove) a BASIC environment file, purge an EW volume, rename an EW volume and rename a BASIC file. The functions of the System Backup Routines are described below.

Backup Entire msus

The Backup entire msus function allows you to copy the entire contents of a mass storage media to another mass storage media. The media is identified by the msus of the mass storage device containing the media. All files (BASIC and EW environment) on the source media are copied. If the destination medium does not have enough room to contain the backup, the system will backup as much as possible. It then issues a message stating that the destination medium is full. You may then insert a new medium in the mass storage device and continue the backup or you may terminate the backup.

Insert an initialized mass storage medium into the mass storage device to which you are sending the backup.

Selecting Backup entire msus causes the system to display all msus currently defined in the configuration file. It then prompts you for the **source** msus. Using the up/down arrow key and the CONTINUE key, select the msus containing the media to be backed up.

Next the system prompts you for the **destination** msus. This is the msus of the mass storage device to which you wish to copy the source. Again using the up/down arrow keys and the CONTINUE key, select the destination msus.

The system then displays the msus of both the source and the destination mass storage device and asks you if they are correct. Responding that the msus displayed represent the correct msus causes the system to copy the entire contents of the source msus to the destinaton msus. If you indicate that either msus is incorrect, the system again prompts you for the source and destination msus.

Copy EWE Volume

The Copy EWE Volume function allows you to backup any volume currently recognized in the system configuration file. All files within the specified source volume are copied to the specified destination volume.

Selecting Copy EWE Volume causes the system to display all volumes currently recognized in the system configuration file. The system then prompts you for to select the volume which you wish to backup. Using the up/down arrow keys and the CONTINUE key, select the volume to be backed up.

Next the system displays the msus of all mass storage devices currently recognized in the system configuration file. It then prompts you for the msus of the device to contain the volume backup. Using the up/down arrow keys and the CONTINUE key, select the appropriate msus. Make sure that an initialized medium is inserted in the mass storage device. The destination media must have enough storage space to contain the entire volume backup.

The system then prompts you for the volume name you wish to assign to the volume backup. A maximum of six characters may be entered, while the first two characters must be the letters, EW. Type in the volume name you wish to assign and press the CONTINUE key. The system next displays a listing identifying the source and destination information for the backup and asks you if the information is correct. You may correct this information at this time. Responding that information is correct causes the system to produce a backup of the volume specified on the destination mass storage device specified.

Copy BASIC File

The Copy BASIC File function allows you to backup any BASIC environment file. The file must be contained in a mass storage device currently recognized by the system.

Selecting Copy BASIC File causes the system to prompt you for the file name of the BASIC file to be backed up. Appending the msus of the device containing the BASIC file to the file name entered allows you to specify a device other than the default device. The default mass storage device is the device containing the system software.

For example, responding:

```
OLD:F8
```

specifies that you wish to backup the BASIC file OLD which is contained on an HP 9885 disc located at select code 8.

Entering no response causes the system to abort the Copy BASIC File function and exit the Copy BASIC file function. If a response is entered, the system verifies that the file is present on the mass storage device specified. Next the system prompts you for the file name of the backup file. Again, appending the msus of the device to contain the backup to the file name allows you to specify a device other than the default device.

If no response is made the system again prompts you for the BASIC file to be copied.

If a file name is entered, the system checks to see if the mass storage device specified has enough room to contain the backup file. This device must be on line and must be part of the system configuration. If there is sufficient room on the mass storage media, the backup proceeds.

Purge BASIC File

This function allows you to remove (purge) any BASIC environment file from any mass storage device. Selecting Purge BASIC File causes the system to prompt you for the name of the BASIC file to be purged and the msus of the device containing that file. Appending the msus to the file name allows you to specify a device other than the default device. The default mass storage device is the device containing the system software. For example, responding:

```
GARB:F8
```

tells the system to purge the file named GARB which is stored on an HP 9885 located at select code 8 (msus = :F8).

The system then displays the name of the BASIC file and the msus of the mass storage device containing the file to be removed. The system prompts you to determine if the information is correct. Answering yes causes the system to remove the BASIC file. Answering no causes the system to again prompt you for the name of the file to be purged and the device containing the file.

Purge EWE Volume

The Purge EWE Volume function allows you to remove any EWE volume from a mass storage media. Once a volume has been purged, the files contained within the volume may never be accessed again. A volume should be purged only after copies of all useful files in the volume have been made. The EWE Filer allows you to make copies of individual EWE files.

Selecting Purge EWE Volume causes the system to display a listing of all volumes currently on line and recognized in the system configuration file. Using the up/down arrow keys and the CONTINUE key, select the volume to be purged. The system then displays the selected volume name and asks you if this is the correct volume to purge. Responding yes causes the volume to be purged. Responding no causes the system to again prompt you for the volume to be purged.

Rename EWE Volume

The Rename EWE Volume function allows you to change the name of any EWE volume currently on line and recognized in the system configuration file. Selecting Rename EWE Volume causes the system to display a list of all volumes on line and recognized in the system configuration file. Using the up/down arrow keys and the CONTINUE key, select the volume to be renamed.

The system next prompts you for the new volume name. Volume names must begin with the letters EW and may be up to six characters in length. Enter the new volume name and press CONTINUE . Next the system displays the old volume name and the new volume name. It then prompts you to determine if the information is correct. An affirmative response causes the system to change the volume name to the name specified. A negative response causes the system to again prompt you for the volume name you wish to change.

Rename BASIC File

The Rename BASIC File function allows you to change the name of a BASIC environment file. The file must be located on a mass storage device currently recognized in the system configuration file. Selecting Rename BASIC File causes the system to prompt you for the name of the BASIC file you wish to rename and the msus of the device containing that file.

Appending the msus to the file name allows you to specify a device other than the default device. The default mass storage device is the device containing the system software. For example, responding:

```
OLD:F8
```

identifies a BASIC file named OLD which is stored on an HP 9885 disc located at select code 8.

Next the system prompts you for the new file name, which may consist of up to six characters. Finally, the system displays the information entered and prompts you to indicate if the information is correct. Answering yes causes the system to change the name of the BASIC file to the name specified. Answering no causes the system to again prompt you for the name of the BASIC file you wish to change.

7908 Backup Utility

The 7908 Backup Utility function allows you to access the backup utility used for tape cartridge backup on the 7908 disc drive. This branch will only appear on EGS/45 systems which have the utility called "BU7908" on their 7908.

Note

The 7908 backup utility comes named "Backup". For EGS/45, this name must be changed to "BU7908". A recommended procedure is to copy the program "Backup" to "BU7908", in order to maintain a utiltiy called "Backup" for operation outside of the EGS environment.

If you do not have a 7908 or you do not have the program "backup" on your disc, this branch will not appear and as a result, you may not select this function.

Selecting this function will cause the utility to be loaded and run. Please refer to the 7908 Backup Utility manual for correct operating instructions. When the 7908 backup utility is terminated, the EGS/45 system will return to the "Backup" menu from which the 7908 utility was loaded.

Note

If you run the 7908 Backup Utility by itself (not from EGS/45), the program will terminate normally.

Shared Resource Management File Utility Program

Introduction

The purpose of this section is to familiarize you with the Shared Resource Management File Utility Program. First, an overview is provided to help you understand the purpose of this program and the way it interacts with the rest of the EGS-45 software. Next, some basic concepts and terms used in this program are discussed. Finally, a detailed description of each function provided by this utility is presented.

Overview

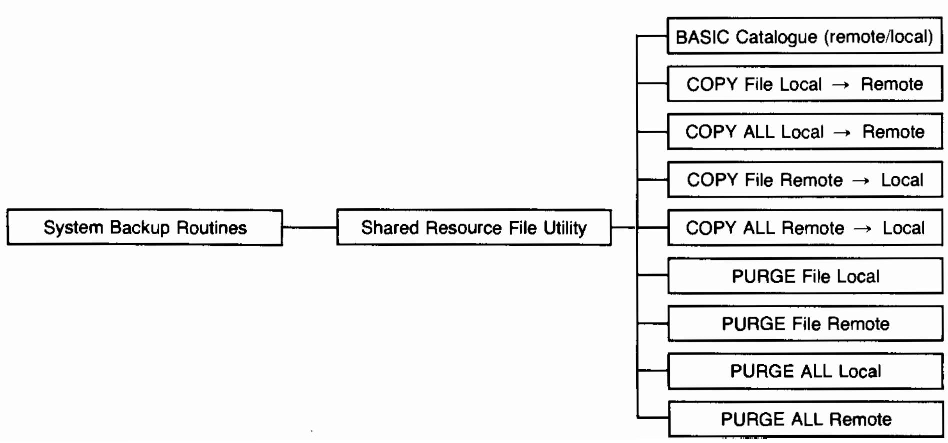
The EGS-45 system operates in two different environments: The BASIC environment and the EW (Editor Workstation) environment as outlined in Chapter 2 of this manual. Each environment has different functions and capabilities. Recall that programs within the BASIC environment cannot directly access files within the EW environment. Rather, the user must first move a file from the EW environment to the BASIC environment. A utility program is provided for this purpose.

In a similar vein, programs within the EW environment cannot directly access files stored on a shared resources disc. These programs can only access files that are stored on a local disc. The purpose of this utility is to transfer files (and EW volumes) between local and shared mass storage. With this program you can:

- copy files from a shared disc to your local disc, thus allowing access to common files such as parts libraries
- copy files from a local disc to a shared disc, thus allowing others to access your work.

There are additional capabilities in the program that are to be discussed in a later section.

The integration of this program with other programs in the EGS-45 system is shown in the following diagram. Like the 7908 Backup Utility Program, this program is selected from the System Backup Routines provided within the File Utilities module. Two steps are necessary to run this program. First, select the System Backup Routine. Then, select the Shared Resource Management File Utility. When the program is terminated, control is returned to the System Backup Routines.

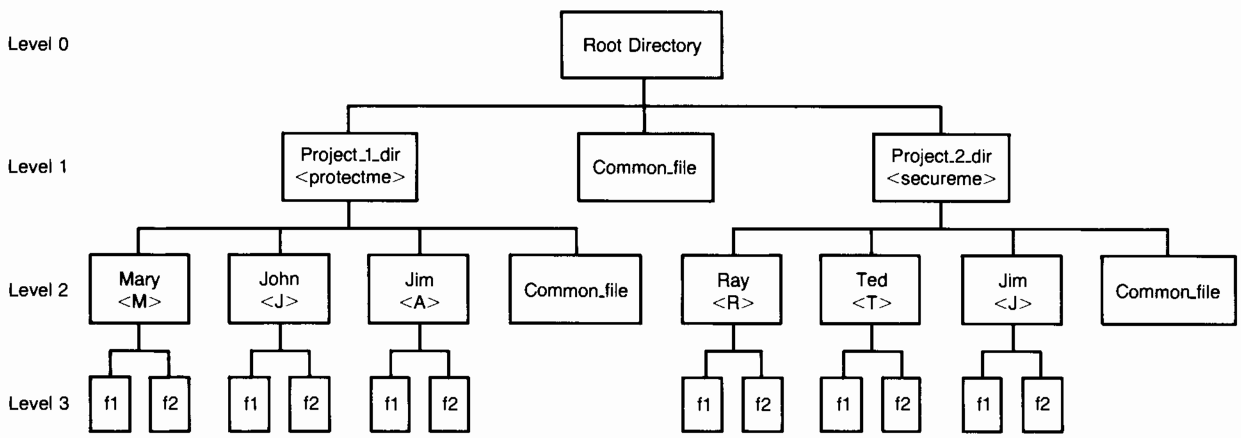


Concepts

The Shared Resource Management System is designed to allow multiple desktop computers to link to a common mass storage system. This allows many users to conveniently share files, yet still have a 'personal mass storage'. A complete description can be found in the manuals supplied with your Shared Resource Management System. The following is included as a brief overview of the system as it relates to this utility program.

Hierarchical Directories

The shared disc in the system contains a hierarchical directory. Unlike a local disc which contains only files, a shared mass storage disc can contain both files and directories. These directories can, in turn, themselves contain additional files and directories to any level of nesting. Typically, this results in a 'tree' structure for the mass storage. An example of this is shown in the following diagram.



Note: Passwords are shown in <_____> form.

Note that it is both legal and reasonable to have duplicate names on a shared disc. The only restriction is that they exist in different directories.

Root Directory

Each remote disc has a root directory which is the highest directory on that disc. All other files and directories are subordinate to the root directory. While the root directory does not have a name, it may be necessary to specify the password of the root directory during the entry of the pathname. To enter the password of the root directory enter '.<root password>' as the first (or only) entry in the pathname.

Passwords

Passwords provide a means of creating a 'personal mass storage' area on the shared disc. With passwords, the user can limit access to his files. Files so protected are as personal as if they were on a floppy disc stored in a desk drawer. Additionally, the system offers various levels of file protection. You can, for example, create public libraries that anyone can use, but only users with the password can change. Passwords contain from 1 to 16 characters (any character except '>') and are preceded by a '<' and followed by a '>'. In the example shown above, the password <secureme> is a password for the directory 'Project_2_dir'.

Pathname

Since there are in general numerous directories on a shared disc, you must define which directory to use in file operations. A pathname is used for this purpose. A pathname is simply a sequential list of directory names, beginning at level 0 (the root directory), which describes how to get to the 'working' directory. In the structure shown above, the pathname:

Project_1_dir<protectme>/Mary<M>

would give access to the two files f1 and f2. Note that successive directory names are separated by '/' and that passwords (enclosed by '<' and '>') are included with each name.

Working Directory

The working directory is the directory specified by the pathname and the remote MSUS. All operations (copy, purge, etc.) take place in the working directory.

Reference Material

For additional information on Shared Resource Management capabilities, refer to the 9845B/C Shared Resource Management Programming Supplement.

Program Entries

There are various entries that a user must make in the Shared Resource Management File Utility Program. These are detailed below:

- The MSUS of the remote disc

  This specifies the shared disc to use for remote file operations.

  The initial default value is:
  :REMOTE

  And this will be the first entry of the MSUS if you elect to explicitly specify the shared resource disc. The subsequent default is of the form:

  :REMOTE xx,yy ; LABEL vvvv<pppp> where:

  xx is requestor select code

  yy is the server node address

  vvvv is the volume name of the remote disc

  pppp is the volume password

  The program automatically supplies these values (except the password) if you do not.

- The pathname to the working directory

  This program assumes that the pathname begins in the root directory. Passwords, if required, are entered with the pathname. A null string means that the working directory is the root directory. If a password is required for the root directory, it it included by specifying '.<root password>' as the first (or only) entry in the pathname.

When copying files to the shared resource disc, the last directory specified in the pathname need not be defined. If it is not, the program gives you the option of creating the new directory or re-entering the pathname.

- The password for the file

This program assumes that the password specified for the working directory applies equally to every file within that directory. The program does not allow you to specify individual password for each file. Rather, as each file is copied to the remote disc, it is protected with the password of the working directory. Manager capability is assigned to the password. Similarly, when copying files from a remote disc to a local disc, the password entered for the working directory must provide read capability. For example, if you specify the pathname:

Project_1_dir<protectme>/Mary<M>

as the path to the working directory, all files copied to the remote disc are placed in level 3 and protected with the password 'M'.

If you chose the option to copy local files to a shared disc and specify the pathname:

EGS_45<secret>

a new directory at level 1 is created and all files placed in that directory are with the password 'secret'. If no password is specified, no protection is provided to the files.

- The remote file name

Remote directory and file names consist of 1 to 16 characters from the following sets: 'a-z', 'A-Z', '0-9', the underscore (_), the period (.) and the foreign characters (ASCII 161-222).

Note that the names of remote files are more restricted than the names of local files even though they contain more characters. This is important to the EGS-45 user as several system files have names that are illegal on the shared disc. Specifically:

CONTROL FILES

These files begin with the character CHR$(1) [CONTROL A]

PARTS FILES (schematic drawing module only)

These files start with the character CHR$(2) [CONTROL B]

Should you desire to copy these files to a remote disc, they must be renamed during the transfer process or an error will result. Similarly, if they are re-copied to a local disc, they must be renamed to include the leading control character or the system will fail to recognize them as special files.

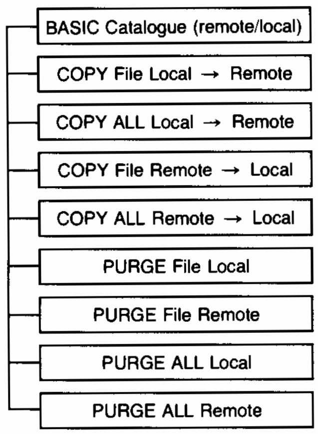
- The local file name and local MSUS

Local file names contain from 1 to 6 characters and may contain any character except CHR$(0), ' ' and ' : '. Local MSUS begin with a ' : ' and a single letter followed, optionally, by the select code and HPIB address. Example entries are ' :F ', ' :Q10,0 '.

If a local MSUS is not specified, a default value is used. When the program is first loaded, the default is the system MSUS. If any other MSUS is entered, it becomes the default for all subsequent entries. The local MSUS is always displayed with the local file name before any operation so that the user may confirm which disc will be used.

Program Operation

After the Shared Resource Management File Utility Program is loaded, the following menu appears. This is the main menu of the program and all options of the program return here upon completion. You may return to the main menu at any time by pressing the CLEAR LINE and CONTINUE keys simultaneously. The only exception to this rule occurs during the entry of the pathname. A null pathname is allowed and will not, by itself, automatically return you to the main menu.



You are prompted to select one of the options using the arrow keys or the cursor. This is exactly the same menu process selection used in other portions of the system.

Catalog a Disc

A menu of the remote disc and all local discs configured in the system is displayed. If you select one of the local discs, it is cataloged and a listing is printed on the system printer. If you select to catalog a remote disc, the program prompts the user to enter the MSUS of the remote disc. The program next prompts you to enter the pathname to the working diectory. The remote disc is thenm cataloged. When the catalog is complete, you can choose to catalog another device or return to the main menu of the program.

Copy a File From a Local Disc To a Remote Disc

REMOTE MSUS: Enter the MSUS of the remote disc. If the entry is not valid, an error message is displayed and the step is repeated.

PATHNAME: Enter the pathname to the working directory. A null entry specifies the root directory. The program next checks the pathname to determine if the working directory exists. If it does not, you have the option of creating it. To change pathname, enter 'N'.

LOCAL FILE: Enter the name and optionally the MSUS of the local file. If the specified file does not exist, an error message is displayed and the step is repeated. A null entry returns the program to the main menu. Be sure to delete the leading control character from control and parts files names.

REMOTE FILE: Enter the name of the remote file. The default is the local file name. If the name is not valid, the step is repeated. The program then checks to see if the file currently exists on the remote disc. If it does, you have the option of purging the existing remote file. A response of 'N' allows you to enter a new remote file name.

CONFIRM: To perform the copy, enter 'Y'. The copy is made and the remote file protected with the password of the working directory. Any errors are reported. The program then returns to the third step so that additional files may be copied.

Copy an Entire Local Disc To a Directory Of The Remote Disc

LOCAL MSUS: Select the source disc from the displayed menu of local discs. If the selected device is not on line, an error message is displayed and the step is repeated.

REMOTE MSUS: Enter the MSUS of the remote disc. If the entry is not valid, an error message is displayed and the step repeated.

PATHNAME: Enter pathname to the working directory. A null entry specifies the root directory.

The program next checks the pathname to determine if the working directory exists. If it does not, you have the option of creating it. To change the pathname, enter 'N'.

CONFIRM: To perform the copy, enter 'Y'. Each file of the local disc is copied to the working directory of the remote disc and protected with the password of the working directory. If an illegal or duplicate name is encountered, the program prints an error message and continues with the next file. When the procedure is complete, the program returns to the main menu.

Copy a File From a Remote Disc To a Local Disc

REMOTE MSUS: Enter the MSUS of the remote disc. If the entry is not valid, an error message is displayed and the step is repeated.

PATHNAME: Enter the pathname to the working directory. A null entry specifies the root directory. The program checks the pathname to determine if the working directory exists. If it does not, an error message is displayed and the step is repeated.

REMOTE FILE: Enter the name of the remote file. If the specified file does not exist, an error message is displayed and the step is repeated. A null entry returns the program to the main menu.

LOCAL FILE: Enter the name and optionally the MSUS of the local file. The default name is the remote name truncated, if necessary, to the first six characters. Be sure and include the leading control character in control and parts file names.

The program then checks to see if the file currently exists on the local disc. If it does, you have the option to purge the existing local file. A response of 'N' allows you to enter a new local file name.

CONFIRM: To perform the copy, enter 'Y'. Any errors are reported. The program then returns to the third step so that additional files may be copied.

Copy an Entire Directory Of a Remote Disc To a Local Disc

LOCAL MSUS: Select the destination disc from the displayed menu of local discs. If the selected device is not on line, an error message is displayed and the step is repeated.

REMOTE MSUS: Enter the MSUS of the remote disc. If the entry is not valid, an error message is displayed and the step is repeated.

PATHNAME: Enter the pathname to the working directory. A null entry specifies the root directory. The program checks the pathname to determine if the working directory exists. If it does not, an error message is printed and the step is repeated.

CONFIRM: To perform the copy, enter 'Y'. Each file within the working directory is copied to the local disc. No directories or files within sub-directories are copied. If duplicate files are encountered, an error message is printed and the program continues with the next file. Should the destination disc become full, you are given the opportunity to insert a new disc and continue copying or abort the process. When the procedure is complete, the program returns to the main menu.

Purge a Local File

LOCAL FILE: Enter the name and optionally the MSUS of the local file. If the specified file does not exist, an error message is displayed and the step is repeated. A null entry returns the program to the main menu.

CONFIRM: To purge the file, enter 'Y'. The file is purged and any errors are reported. The program then returns to the first step so that additional files may be purged.

Purge a Remote File

REMOTE MSUS: Enter the MSUS of the remote disc. If the entry is not valid, an error message is displayed and the step is repeated.

PATHNAME: Enter the pathname to the working directory. A null entry specifies the root directory. The program checks the pathname to determine if the specified directory exists. If it does not, an error message is displayed and the step is repeated.

REMOTE FILE: Enter the name of the remote file. If the specified file does not exist, an error message is displayed and the step is repeated. A null entry returns the program to the main menu.

CONFIRM: To purge the file, enter 'Y'. The file is purged and any errors are reported. The program then returns to the third step so that additional files may be purged.

Purge an Entire Local Disc

LOCAL MSUS: Select the MSUS of the local disc from the menu of local discs. If the selected device is not on line, an error message is displayed and the step is repeated.

CONFIRM: To purge the entire local disc, enter 'Y' twice. Each file on the local disc is purged. Any errors are reported. When the operation is complete, the program returns to the main menu.

Purge an Entire Directory of a Remote Disc

REMOTE MSUS: Enter the MSUS of the remote disc. If the entry is not valid, an error message is displayed and the step is repeated.

PATHNAME: Enter the pathname to the working directory. A null entry specifies the root directory. The program checks the pathname to determine if the specified directory exists. If it does not, an error message is displayed and the step is repeated.

CONFIRM: To purge the directory, enter 'Y' twice. Each file within the working directory is purged. Any errors are reported. The program also attempts to purge the working directory. When the operation is complete, the program returns to the main menu.

Table of Contents

Chapter 4
System Configuration Module

Introduction

The System Configuration Module allows you to modify the system configuration. The system configuration tells the system which peripherals (such as printers, paper tape punches, and mass storage devices) are connected to the system. Additionally, it tells the system the select code (and address, if applicable) of each device. Changes made to the system configuration may be stored in a file on your mass storage media, called the system's configuration file. This file is loaded into the system each time the EGS-45 software is loaded and run.

When the system configuration is modified, the changes are made to a copy of the system's configuration file in computer memory. This means that these changes are valid only until the system is turned off. You may make these changes to the system configuration permanent by saving them in the system configuration file.

This module allows you to add and delete peripheral devices, change the address and select code of the peripheral devices, and create new EWE volumes in which to store your EWE files. Before modifying the system configuration, connect the new peripheral to the system and turn the power on. The system attempts to verify any new additions to the configuration at the time the change is made. If the new peripheral is not found at the location specified, it will not allow you to modify the configuration.

The system diagram (located at the beginning of Chapter 1) shows the options available with the Configuration Module. Selecting one of these options (Change Peripheral, Change Mass Storage, etc.,) allows you to identify the portion of the system configuration that you wish to modify. The system then displays a subset of options which allow you to make specific changes to the system configuration.

Change Mass Storage

Selecting Change Mass Storage allows you to modify the system configuration by:

- adding a mass storage device
- deleting mass storage device
- changing the select code (and address, if applicable) of a device
- saving the new mass storage configuration in the system's configuration file

Additionally, it allows you to obtain a printed table of all mass storage devices in the system configuration.

When you are finished modifying the system configuration, exit the Change Mass Storage option by selecting Return to Manager . The system then automatically checks to see if any changes have been made to the configuration of mass storage devices. If the configuration has been changed, the system checks to see what EWE volumes are present in these devices. It then displays a table of these volumes on the system CRT. You may obtain a printed copy of this table by pressing the special function key (K0).

Selecting Change Mass Storage causes a new menu of choices to be displayed. These choices are detailed below.

ADD Mass Storage
Selecting ADD Mass Storage causes the system to prompt you for the mass storage unit specifier (msus) of the new mass storage device. Simply type in the new msus and press (CONTINUE). The system then checks to see if the device is connected to the system. If the device is found at the location specified, it is added to the system configuration. If the device is not found, you are asked to re-enter the msus of the device. When you are finished adding mass storage devices, make no entry in response to the prompt for the device msus (press only (CONTINUE)). This causes the Change Mass Storage menu to be redisplayed.

Delete Mass Storage
Selecting Delete Mass Storage causes a list of the msus of all mass storage devices in the system configuration to be displayed. Entering the msus of the device you wish to delete, causes the system to prompt:

```
Are you sure you wish to remove msus?
```

Answering Y, for yes, causes the device to be removed. The system then prompts you for additional devices to be deleted. Answering N, for no, causes the system to again prompt you for the msus of the device to be deleted.

When you are finished deleting mass storage devices, make no entry in response to the prompt for the device msus. This causes the Change Mass Storage menu to be redisplayed.

List Configuration
Selecting List Configuration causes a table of all mass storage devices in the system configuration to be displayed. Pressing the special function key, K0, causes the table to be copied to the computer's built-in printer. Pressing (CONTINUE) causes the Change Mass Storage menu to be displayed again.

Save Configuration
Selecting Save Configuration causes the system to replace the mass storage device portion of the system configuration file with the current mass storage device configuration. The system configuration file is the file that tells the Manager which devices are connected to the system at the time EGS-45 is loaded and run.

Before saving the new configuration, the system asks you if you really wish to save the configuration. Responding Y, for yes causes the configuration to be saved. Responding N, for no, causes no changes to be made to the system's configuration file. If the configuration is not saved, changes made to the configuration with Change Mass Storage are active until EGS-45 is terminated and then are lost.

Change Peripherals

The Change Peripherals branch allows you to:

- add peripheral devices to the system configuration
- delete peripheral devices from the system configuration
- obtain a listing of the peripherals in the system configuration
- change the select code and/or address of a device in the system configuration
- save the current peripheral configuration in the system's configuration file

Peripherals (other than mass storage devices) in the system configuration are organized into a peripherals table. Each device in the table has a number, a device type (a name that you assign to the device), a select code, and an address (if applicable). The system reserves the first three slots in the peripheral table as follows:

#1 is reserved for the system graphic input device (either the HP 9874 or the HP 9111A/T)
#2 is reserved for the system plotter
#3 is reserved for the system printer

Slots #9 and #10 of the peripheral table are used by the P.C. Board Layout module for the HP 9884 paper tape punch and the HP 7970E magnetic tape drive, respectively.

The remaining slots in the table may be assigned as you need them by selecting Add Peripheral, Delete Peripheral, or Change Select Code.

Unless you save the system configuration, changes made to the peripherals table are active only until EGS-45 terminates and then are lost. Saving the current configuration causes the system to replace the peripherals table portion of the system configuration file with the current peripherals table.

SYSTEM PERIPHERALS

	TYPE	SELECT	ADDRESS
#1	9111A	7	6
#2	NO PLOTTER	7	5
#3	PRINTER	0	
#8	PAPER PUNC	2	
#9	7970E MAG	12	1

Add Peripheral

Selecting Add Peripheral allows you to add a peripheral device to the system configuration. This allows the device to be accessed by the EGS-45 software. When Add Peripheral is selected, the system displays the current peripherals table. It then prompts you for the slot number of the peripheral table where the new device is to be listed. Entering no response to this prompt causes the Change Peripheral menu to again be displayed.

For example, responding with the number 5 allows you to add a new peripheral to slot number 5 of the peripheral table. If a slot is already occupied, you must first remove the existing device from the table by selecting Delete Peripheral from the Change Peripheral menu.

Next the system asks you for the type of peripheral you are adding. This is simply a 20 character label identifying the device in the table. The response you supply is displayed beside the peripheral number in the peripheral table. You should repsond with a combination of characters that helps you identify the device. For example, if the device is an HP 9872C pen plotter, you might respond, `HP 9872 Plotter`. This causes `HP 9872 Plotter` to appear in the peripheral table adjacent to the slot number of the device. If you do not supply a peripheral type, the system again prompts you for the slot number of the device that you wish to add to the table.

The system then asks you for the select code of the interface connecting the peripheral to the computer. Enter the appropriate select code and press (CONTINUE). If no select code is entered, the system again prompts you for the number of the device you wish to add to the peripheral table.

The system next prompts you to determine if the peripheral has an address. Only devices connected to the computer with an HP-IB interface have an address. If the device has an address, answer yes to the prompt. Then supply the correct address of the device when prompted by the system.

Delete Peripheral

Selecting the Delete Peripheral function allows you to remove any peripheral from the current configuration. Once selected, the system displays the current peripheral table. The system then prompts you for the slot number of the peripheral you wish to remove from the table. Making no response to this prompt and pressing (CONTINUE) causes the system to assume that you do not wish to delete a device from the table. The Change Peripheral menu is then displayed. If you supply a slot number from the table, the system asks you if you are sure you wish to remove this device from the configuration. Responding yes causes the device to be removed.

Change Select Code

Selecting Change Select Code allows you to change the select code and/or address of any device already present in the peripheral table. This is extremely useful for quick or temporary changes to the system configuration. Selecting this function causes the system to display the current peripherals table.

The system then prompts you for the slot number of the device whose select code and/or address you wish to change. If no response is entered and the CONTINUE key pressed, the Change Peripheral menu is again displayed. If a slot number from the table is entered, the system then prompts you for the new select code of the device. If no select code is entered and the CONTINUE key is pressed, the system again prompts you for the slot number of the device whose select code you wish to change.

Next the system asks you if the device has an address. Only devices connected to the computer with the HP-IB interface have addresses. Responding no causes the system to update the peripheral table (and thus, system configuration) to include the new select code. Responding yes causes the system to prompt you for the device address. Entering an address causes the new select code and address of the device to be placed in the peripheral table. If no address is supplied, the system again prompts you for the slot number of the device whose select code/ address you wish to change.

Save Configuration

Selecting Save Configuration causes the system to replace the peripherals table portion of the system configuration file with the current peripherals table. The system configuration file is the file that tells the Manager what devices to look for when the software is loaded and run. If the configuration is not saved, changes made to the peripheral table are active until EGS-45 is terminated and then are lost.

List Configuration

Selecting the List Configuration function causes the system to display the current peripherals table. Pressing the special function key (k0) causes the table to be copied to the computer's built-in printer.

Add EW Environment Volume

Selecting Add EW Environment Volume allows you to create an Editor Workstation environment volume. An EWE volume is a BASIC data file used by EGS-45 to store EWE files (such as device files, macro files, amd process files). Volumes may be created on any supported mass storage device. Before an EWE volume may be created, the medium in the mass storage device must first be initialized with the BASIC language INITIALIZE statement (see your computer's Operating and Programming manual). Initialization of the media prepares it for use by your computer and need be performed only once in the media's lifetime.

Note

Initializing mass storage media erases all programs and data stored on the media. Do not initialize any media containing information that you wish to retain.

Selecting Add EW Environment causes the system to display a menu of all current mass storage devices (identified by their msus). It then prompts you to select the device that is to contain the new volume. Once you have selected the mass storage device, the system prompts you to enter the new volume name. EWE volume names are composed of up to six characters. The first two characters of the volume name must be EW. For example, EWMINE, EWTOM, EWDRAW, and EW1 are all legal volume names. The characters EW allow the system to recognize the BASIC environment data file as an EW environment volume. Entering no response to this prompt (clearing the line of any characters and pressing (CONTINUE)) causes the system to again display a list of all current mass storage devices. You may then reselect the device to contain the new volume or return to the System Configuration menu.

After entering the new volume name, the system prompts you to enter the size of the new volume, in records. The size of the volume depends upon the amount of data to be stored in the volume. Determining the size of the volume is largely a matter of experience. The minimum volume size is 300 records. If no volume size is entered, a list of all current mass storage devices is again displayed. You may then select the device to contain the new volume or return to the System Configuration menu.

Reconfigure EW Environment

Selecting Reconfigure EW Environment causes the system to check all mass storage devices in the current system configuration and identify all EWE volumes on line. This branch must be selected whenever you introduce a new volume to the system. A new volume is introduced to the system when you insert a disc into a mass storage device after system start-up or after the last reconfiguration. A listing of all volumes identified is printed on the system printer and the System Configuration menu appears again.

Table of Contents

Chapter 5
Command File Facility

The EGS-45 software allows you to create, store, modify, and plot drawings by using the commands of Graphics Editor and its subsystems. These commands are entered interactively, one at a time, to achieve the desired results. Occasionally, you may know the series of commands that you want to enter. It would be advantageous if you could get the computer to execute these commands without you being there to enter them interactively. For example, you may desire to create several generate files for use by one of the application modules. While only one or two commands are necessary to create each file, the time required is rather lengthy. You could find more productive uses for your time than waiting to enter each command interactively.

The Command File Facility allows you enter many Graphic Editor commands (and commands of the P.C. Board Layout module) and store them in a special file called a command file. A command file is an EWE file that is stored in an EWE volume by the Command File Facility. When the command file is executed, the system reads and executes each command in the command file in the order that they are entered. Because the system executes each command without interaction and continues until all commands are executed, it is essential that no errors are made in creating the command file. This also means that all prompts by the system must be anticipated and answered.

The Command File Facility gives access to two options: Edit/Create Command File and Execute Command File. Each of these options is discussed in the sections that follow.

Edit/Create Command File

Concepts

Selecting the Edit/Create Command File branch causes the system to prompt you for the name of the command file to be edited. If you wish to edit an existing command file, enter the file name of the command file. If the file exists on a volume other than EWSYS, you must precede the file name with the volume name and a colon. For example, to edit the command file ABC that resides on the volume EW1, you would enter EW1:ABC. If you are creating a new command file, simply press (CONTINUE) or (STORE) (the system automatically selects the Add function for you).

Next the system displays a set of functions that you may select:

A)dd Q)uit C)lear I)nsert D)elete R)eplace V)erify H)elp

These functions allow you to create and/or modify the command file. To select one of the functions, type the first letter of the function. For example, typing the letter A causes the A)dd function to be selected. These functions are discussed in a later section.

The command file is an EWE file stored on your mass storage medium by the Command File Facility. A command file is nothing more than a series of commands (from the Graphics Editor, any of its subsystems, or the P.C. Board Layout module) that are executed sequentially from the beginning of the command file to its end. Since these commands are executed unconditionally, care must be taken to insure that all commands are correct before attempting to execute the command file. If the command generates a prompt by the system when entered interactively, the prompt will also result when the command is executed from the command file. Therefore, you must anticipate these prompts and include the proper response in the command file.

The command file consists of up to 20 lines of commands with a maximum of 75 characters per line of the file. This allows several Graphics Editor commands to be entered on one line. Additionally, a command file can call, or reference another command file. In this fashion, many command files may be chained and an unlimited number of commands executed.

Before the Graphics Editor and/or its subsystems may be accessed, the following command file line must be entered:

`XGEDIT.CODE` (CONTINUE) or (STORE)

This causes the Graphics Editor to be loaded into the system. Before using the edit subsystem, you must first explicitly load the process and macro files you intend to use in the edit subsystem. The command file line below accomplishes this task:

`PRO vol:filename;EXIT;MAC vol:filename;EXIT;` (CONTINUE) or (STORE)

where `vol` is the volume name containing the process/macro file and `filename` is the name of the process/macro file. The default volume is the system volume, EWSYS.

If you wish to access the P.C. Board Layout module from a command file, other special command file lines must be entered. These command file lines are discussed in detail in the P.C. Board Layout module chapter.

To execute another command file from within a command file, the following command file line must be entered:

`Cvol:filename` (CONTINUE) or (STORE)

where `vol` is the volume name on which the command file is stored and `filename` is the name of the command file. If a command file is called by your command file, the call (Cvol-:filename) must be the last line of the command file.

Once the Graphics Editor has been loaded into the system, entering commands in the command file is identical to entering them interactively in the Graphics Editor. To illustrate, the command file shown below causes the drawing EXAMPLE to be loaded into the edit subsystem from the volume EW1, fitted onto the screen, and copied to the built-in printer via the DUMP command.

If the command file calls the graphics editor and goes into the EDIT subsystem, the program will pause. To prevent this problem, there are two solutions. One is to simply press a space when the system pauses; the other is to make sure the tablet is off. Both fixes will cause the command file to be executed normally and without problem.

```
1 XGEDIT.CODE
2 ECHO;
3 PRO PROCDATA; EXIT; MAC MACRDATA; EXIT;
4 EDIT EW1:EXAMPLE; SHOW #E; WINDOW :F; DUMP;
5 EXIT;
```

The first line of the command file above loads the Graphics Editor into the system. The second line of the command file turns on echo. This allows you to have a list of all commands entered by the command file and any messages issued by the system. Until you feel extremely comfortable writing command files, turning echo on with the command file is highly recommended.

The third line of the command file loads the process file PROCDATA from the default volume (EWSYS), exits the process subsystem, loads the macro file MACRDATA from the default volume, and exits the macro subsytem.

The command file next contains the command `EDIT EW1:EXAMPLE;`. This loads the device file EXAMPLE from the volume EW1 and enters the edit subsystem. Notice the last command in the file is `EXIT`. This means that when the command file is finished executing, the system will display the prompt `MAIN>`, indicating that you are in the Graphics Editor. To completely exit the Graphics Editor, you would add the Graphics Editor command `BYE;` as line 6 in the command file above.

You **must** anticipate the prompts that the system issues in response to commands entered by the command file. Many commands cause the system to generate extra prompts once the command has been invoked. A prime example is the PLOT command, illustrated in the example:

```
1 XGEDIT.CODE
2 ECHO;
3 PROCESS PROCDATA; EXIT; MACRO MACRDATA.MAC; EXIT;
4 EDIT EXAMPLE; SHO#E; PLOT;
5
6 WINDOW :F; DUMP;
7 EXIT;
8 BYE;
```

Note that the STORE or CONTINUE key was pressed twice following the PLOT command. Once following `PLOT;` to tell the system to accept the PLOT command. Pressing CONTINUE or STORE ends a command file line and, when the command file is executed, is equivalent to pressing the key interactively. The second STORE or CONTINUE key associated with the PLOT command ends command file line 5 above. This is in response to the prompt "Place paper in plotter, set limits, press CONTINUE" which is generated when the PLOT command is invoked. This implies that you are sure that there is paper in the plotter and that the plotter limits are set correctly.

Many commands available in Graphics Editor and a few in the edit subsystem generate extra prompts such as that for the PLOT command in the example above. Before attempting to execute a command file, you should try to execute the individual commands interactively to verify what prompts need to be answered in your command file.

Functions of Create/Edit Command File

When modifying an existing command file, a menu of functions is displayed at the top of the CRT. To select a function, simply type the first letter of that function. For example, to select the A(dd function, type A. The capabilities provided by each of these functions is described below.

To obtain a printed copy of the command file while you are creating/editing it, simultaneously press CONTROL and the special function key, k0.

A(dd

The A(dd function allows you to add commands to a command file. Selecting this function causes the system to position the cursor at the start of the next new line in the command file (line one if it is a new command file). You may then begin typing in the EGS-45 commands that compose your command file. These commands should be entered in the order that you want them executed.

New lines are obtained by pressing either CONTINUE or STORE. This is equivalent to pressing either key when the commands are entered interactively.

Pressing SHIFT and PAUSE (simultaneously), followed by CONTINUE at the beginning of a new line causes the system to exit the A(dd mode.

C(lear

Selecting the C(lear function causes the system to delete all lines from the current command file.

I(nsert

The I(nsert function allows you to insert a new command file line into the existing command file. Selecting this function causes the system to display the prompt:

```
INSERT NEW LINE AFTER LINE NUMBER ==>
```

Enter the line number of the command file after which the new line is to be inserted. Once the line number is supplied, the system renumbers the file so that a blank line is inserted at the location specified. The system positions the cursor at the beginning of this line, thus allowing you to immediately enter the new line.

To insert a new line before line number 1, enter line number 0 in response to the above prompt.

D(elete

The D(elete function allows you to delete an individual line from the command file. Selecting this function causes the system to prompt you for the line number of the command file that you wish to delete. Entering the line number causes the system to delete that line.

R(eplace

The R(eplace function allows you to replace an existing command file line with a new command file line. Selecting this function causes the system to prompt you for the line number that you want to replace. Entering the line number causes the system to delete that individual command file line. It then positions the cursor at the beginning of that line so that you may type in its replacement.

V(erify

Selecting the V(erify function causes the system to redisplay the command file on the system CRT.

H(elp

Selecting the H(elp function causes the system to display a help message for each of the functions of Edit/Create Command File. To obtain a printed copy of this help message, simultaneously press CONTROL and the special function key, k0.

```
A(dd   adds lines after the last line.

Q(uit   exits the editor and optionally outputting to a file.

C(lear  clears all lines.

D(elete  removes a line.

I(nsert  inserts a new line after an existing line.

R(eplace  replaces an existing line with a new line.

V(erify  re-displays the lines.

H(elp  prints this text.
```

Q(uit

Selecting the Q(uit function tells the system that you are finished creating/editing the command file. The system then prompts you to enter the name that you wish to assign to the command file. Entering a file name (15 characters maximum) causes the system to create a command file on the default volume (EWSYS). To store the command file on a different volume, precede the file name with the volume name and a colon. For example, entering EW1:ABC stores the command file with the file name ABC on volume EW1. The system then displays the Manager menu.

Entering no file name causes the system to attempt to save the file with the command file name entered when Edit/Create Command File was selected. If no file name was supplied at that time, you must supply a file name with the Q(uit function if you wish to save the file.

Simultaneously pressing (SHIFT) and (PAUSE), followed by (CONTINUE) causes the system to assume that you do not want to store the command file. The file is not stored and the system then displays the Manager menu.

Changes made to an existing command file are temporary until the command file is resaved with the Q(uit function.

Execute Command File

Selecting Execute Command File causes the system to prompt you for the name of the command file to be executed. To specify a command file that is stored on a volume other than EWSYS, precede the command file name with the volume name containing the file and a colon. For example, entering EW1:SAMPL causes the system to load and execute the command file SAMPL that is stored on EW1.

Commands are executed one by one from the beginning of the command file to the end. You may stop the execution of a command file by simultaneously pressing (SHIFT) and (STOP). The system then issues a prompt that allows you to continue or to stop execution of the command file.

Table of Contents

Chapter 6
User Written Applications

Introduction

This chapter describes how your application programs may be integrated with the modules offered in this system. It discusses how to organize your application program so that they may be selected from the Manager menu, just like any other application module. It also discusses methods of reading EWE files into your BASIC language program(s) so that you may operate on the data contained in the file.

Your BASIC language and/or ASSEMBLY language application programs may be used to:

- Post process drawing data (convert it from its present form to another form)
- Print reports
- Perform analysis
- Perform other special tasks

To allow you to select your application program from the Manager menu, your program must be written with a special structure. In addition, you must create a control file. A control file defines an EGS-45 module and identifies the BASIC language application programs that you wish to access as branches to the module.

This chapter assumes that you have a working knowledge of the BASIC language and are well acquainted with your computer.

Control Files

A control file is a BASIC language program, comprised of comment statements, that is saved on your mass storage medium. It defines and describes your application program(s) and allows these programs to be selected from the Manager menu. The control file is saved on your mass storage medium using a special file name which enables the Manager to recognize it as a control file. The form of the control file is described below along with the procedures for saving the file on mass storage medium.

Structure

EGS-45 requires that control files follow a rather strict structure. The control file must be organized into blocks of program lines. The line identifiers of the BASIC statements forming the control file are ignored. Instead, EGS-45 counts **the number of program lines** to determine what information is contained in the file.

The first program line of the control file (regardless of the line identifier assigned to it) is defined to be the Application Module name. The next seven program lines are defined to be the Help Message for the application module. The ninth program line specifies the total number of BASIC language application programs that are to be selected from this application module. Each application program included in this number must be described in the remaining lines of the control file.

A block of ten BASIC program lines are used to describe each application program. The first program line in each block specifies the application branch name. The next program line in the block must be the name of the BASIC file in which the application program is stored. The remaining eight lines of the set are the help message for the application branch.

The structure of the control file and the contents of each line are represented below.

Line	Program Line Contents
1	! Application module name
2-8	! Help message
9	! Number of branches (application programs)
10	! Application branch name
11	! Application program name
12-19	! Help message
20-??	! lines 10-19 repeated for each additional program included in the number of application programs in line 9)

Where

Line - The line numbers referenced here represent the physical line number of the control file, not the BASIC line identifier associtated with the program line. The physical line number is the number you would find if you counted the numbered lines of the program from the program's start upto and including the program line described.

! - The exclamation mark is used in the BASIC programming language to identify a line as a comment. An exclamation mark is required as the first character in each line of the control file.

Application module name - This is the name of the application module as you want it to appear in the Manager menu. It may consist of a maximum of thirty alphanumeric characters. EGS-45 reads the characters in this line and adds them to the Manager menu as the name of an application module. Selecting this module then allows you to access your BASIC language application programs, which appear as branches to the module.

Help message - The information contained in these program lines should describe the use and operation of the application module. This information is the message that is printed when the special function key (k0) is pressed while the module name is highlighted in the Manager menu. All eight lines must exist whether they are needed for the help message or not.

Number of branches - This is the number of BASIC language application programs that you wish to access from this particular module. Each application program appears as a branch to the module in the Manager menu. Each branch requires that a branch name, a program name, and a help message be defined in the control file.

Application branch name - This is the name that appears in the Manager menu allowing you to select the application program. A maximum of thirty (30) alphanumeric characters may be entered as the branch name. Each application branch name is associated with the name of a BASIC language application program.

Program name - This is the name of the file in which you have stored your BASIC language application program.

Help message - The information contained in these control file lines describes the use and operation of the associated application program. This information is printed when the special function key (k0) is pressed while the branch name is highlighted in the Manager menu. All eight lines of the help message must exist whether they are needed for the help message or not.

The systems Graphics Editor module may be accessed by your application module. This is done by adding a slight modification to the line containing the name of an application program. One of the application programs must be the BASIC language program that is the Graphics Editor. You must then add additional information to the control file line identifying the Graphics Editor. This information acts as a "mini-command file" and is immediately executed when the Graphics Editor branch is selected from your application module.

If the command file calls the graphics editor and goes into the EDIT subsystem, the program will pause. To prevent this problem, there are two solutions. One is to simply press a space when the system pauses; the other is to make sure the tablet is off. Both fixes will cause the command file to be executed normally and without problem.

When creating a control file that accessed the Graphics Editor, you are required to enter a special character at several locations within the "mini-command file". The special character is entered by simultaneously pressing (CONTROL) and (M). This causes a carriage return character to be entered in the mini-command file. You should enter this combination of keys whenever a carriage return is called for in the control file (represented by the characters <CR>). Pressing (CONTROL) (M) causes a special character resembling a miniature letter C with the subscript R to be displayed.

To allow the Graphics Editor to be selected from your application module, the following control file line should be entered as the name of an application program:

```
XXX   !   SYSEWE XGEDIT.CODE <CR>(<CR> = CONTROL M )
```

This tells the system to load the basic file named SYSEWE, which allows access to EW environment programs. The rest of the line (XGEDIT.CODE <CR>) is the mini-command file. When this branch is selected from the Manager menu, the Graphics Editor is loaded and the prompt MAIN> displayed.

The program line above could be expanded to cause a particular process file, macro file, and/or menu file to be loaded. For example, modifying line XXX above:

```
XXX   !    SYSEWE XGEDIT.CODE<CR> PRO WIRE ;<CR> EXI;<CR>
MAC LINES;<CR> EXI;<CR> EDI;<CR> MEN WIREMEN;<CR> EXI;<CR>
```

When the branch is selected from your application module, the Graphics Editor is loaded. The process subsystem is then accessed and the process file WIRE loaded. The process subsystem is exited and the macro susbystem entered, loading the macro file LINES. Finally, the edit subsystem is entered, the menu file WIRMEN loaded, and the edit subsystem exited. In this fashion, you may load specialized process, macro, and menu files with the Graphics editor. This allows complete customization of the Graphics Editor for your application. Up to 160 characters may be entered in this mini-command file.

The special character you entered when adding the Graphics Editor to your control file may be displayed on your system CRT, but cause some difficulty when printed. To allow your control file containing these special characters to be printed, a special BASIC language PRINT statement must first be executed:

```
PRINT CHR$(27)&"Y" EXECUTE
```

This statement causes all special characters to be printed when listing a program to the printer. To disable this feature of printing special characters another BASIC language statement must be executed:

```
PRINT CHR$(27)&"Z" EXECUTE
```

An example of a control file that allows the selection of the Graphics Editor is shown below. The mini-command file causes the process file PC to be loaded from the volume EWPC.

```
10  !  GENERAL EDITING
20  !  GENERAL EDITING COMMAND
30  !
40  !  This branch will allow you to perform general purpose
50  !  editing using the graphics editor.
60  !
70  !
80  !
90  !  1
100 !  GRAPHICS EDITOR
110 !  SYSEWE XGEDIT.CODE <CR> PRO EWPC:PC; EXIT; <CR>
120 !
130 !  This mini-command file executes the graphics editor and loads
140 !  it into memory.  The process file PC is loaded from the volume
150 !  EWPC.
160 !
170 !
180 !
190 !
```

To help understand the structure of the control file, the control file for the Configuration Module is shown below.

```
10    !   System Configuration
20    !   The System Configuration Routines
30    !
40    !     This branch of the main system will allow for the change and
50    !       modification of the current system configuration.  Among the
60    !       tasks which can be performed are:
70    !         * changing the peripherals      * changing the mass storage
80    !         * re-configuration of the EWE   * add EWE volumes
90    !   4
100   !   Change Peripherals
110   !   Cperff
120   !   Change Peripherals
130   !
140   !     This command will allow the user to change and add to the
150   !       peripherals table.  The new table may be stored so that it
160   !       will be used the next time EGS-45 is executed.
170   !
180   !
190   !
200   !   Change Mass Storage
210   !   Cmsus
220   !   Change Mass Storage
230   !
240   !     This command will allow the user to add and change the current
250   !       mass storage device table.  As units are added to the system
260   !       the program will check to see if the device is really there.
270   !       The table may be save perminently if desired.
280   !
290   !
300   !   Re-configure EW Environment
310   !   ConfEW
320   !   Re-configuration of the EW Environment
330   !
340   !     This command will not require any form of input.  Its function is
350   !       to simply check the system for all possible EWE files.  The
360   !       files which are found are put into a table and displayed on the
370   !       screen.  There is one option, that being the dumping of the
380   !       screen to the internal thermal printer.
390   !
400   !   Add EW Environment Volume
410   !   addEWE
420   !   Addition of Editor Workstation Environment files to the system
430   !
440   !     This command will allow for the addition of Editor Workstation
450   !       files to the existing system.
460   !
470   !
480   !
490   !
500   !
```

Saving the Control File

Once the control file has been created, it is necessary to save the program on your mass storage media. The program must be saved using a special file name that enables the Manager to recognize it as a control file. The first character of the control file name is specified by `CHR$(1)`. The remaining five characters may be any alphanumeric characters. The form of the BASIC SAVE statement that you use to save the control file is shown below.

```
SAVE CHR$(1)&"XXXXX" (EXECUTE)
```

where the characters `XXXXX` represent the name of the control file in the mass storage directory. To load the control file into memory so that you may modify it, the following form of the BASIC GET statement is used.

```
GET CHR$(1)&"XXXXX" (EXECUTE)
```

Application Program Structure

To enable each of your application programs to be used by EGS-45, the first nine lines of each program must appear exactly as shown below.

```
10  COM INTEGER System_on,Extra_com(1:93),Crt$[10],Date$[20],Ew_date$[2],INTEGER Memory
20  COM Perif_type$(1:10)[20],INTEGER Perif_select(1:10), Perif_address(1:10), N_perif
30  COM System_msus$[10],Msus$(1:5)[10],INTEGER N_msus
40  COM Ewe_select$(1:50)[10],Ewe_offset(1:50),Ewe_file_name$(1:50)[6],
    INTEGER N_units
50  COM Control_file$(1:12)[16],Tree$(1:12)[32],INTEGER N_applications
60  COM Branches$(1:12,1:6)[32],Branches_name$(1:12,1:6)[6], INTEGER
    N_branches (1:12)
70  COM INTEGER Iv1,Iv2,R,C,Present_menu,Current_command$[32]
80  COM INTEGER Max_characters,Max_lines,Screen_width,Space,Manager$[16]
```

These lines of BASIC code declare variables that are common to the EGS-45 software and to your application program. This is done so that information is not lost when moving back and forth between the Manager and your application program. Your application program may access any of these variables*. A short description of each variable is provided at the end of this section.

To link back to EGS-45 after completion of your application program, the following BASIC line must be appended to the end of your program:

```
XXXX LOAD Manager$
```

where `XXXX` is greater than the last line number of your program.

This redisplays the Manager menu after completion of your program, allowing a different module to be selected.

* To aid you in adding this code to your program, it is saved in a file called COM.

If your application program uses the programming tools, PASSIGN and PREAD, the following lines must be added to the section of BASIC code above:

```
100    INTEGER Error,Lrs,File_type
110    REAL  Npr,Lpr
120    DIM Buf$[2560],Data$[256]
130    IDELETE ALL
140    ILOAD "FDES"&System_msus$
150    ILOAD "MSIO"&System_msus$
```

The PASSIGN and PREAD programs are described later in this chapter.

Common Variable Definition

In the following section, each of the variables required in common at the beginning of your application program are discussed. This allows you to use and modify these values in your program.

`System_on` - INTEGER variable which is set to 0 at startup and set to 1 after the system is running. It is used to prevent loading the start program when returning to EGS-45 from another program.

`Extra_com` - This is an integer array of 1 by 93 which is extra space in the common area. It is present to allow for future expansion of the EGS/45 common.

`Crt$` - STRING of 10 characters which contains the CRT type. The value of this variable is set in the EGS-45 start-up routine. It may have one of the following values:

> `CYCLOPS` - identifies an older version of the black and white CRT
> `AURORA` - identifies new version of the black and white CRT
> `ODYSSEY` - identifies color CRT

`Date$` - STRING of 20 characters which contains the current date. This is the date entered when EGS-45 is loaded and run.

`Ew_date$` - Special date for the EW environment. The date is pack into 2 bytes. This packing is done in a routine called Ew_date, in the start routines.

`Memory` - This is an integer, whose value ranges from 1 to 9. It is read by the start routines and corresponds to a table which indicates the amount of memory in your computer. This variable is set by the AUTOST program.

The following set of array variables form the peripherals table (part of the system configuration). There are a maximum of 10 entries into this table. The first three entries are reserved by the system.

If the string value of any of the 10 array items is NULL (empty) then there is no peripheral in that slot of the peripheral table. You may wish to query these variables to determine the select code and address of any peripheral entered in the peripheral table (such as printers, plotters, and mag tape). Refer to the System Configuration chapter for more information on the peripheral table.

`Perif_type$` - STRING ARRAY containing the peripheral type of each entry in the peripheral table. Each item of the array consists of up to 20 characters.

`Perif_select` - INTEGER ARRAY containing the select code of each entry in the peripheral table.

`Perif_address` - INTEGER ARRAY containing the address of each entry in the peripheral table. These values range from 0 to 31.

`T_on` - INTEGER used as a BOOLEAN to determine if the tablet is to be used by the manager for selection of menu items.

The following variables are similar to the peripheral table variables. They identify the system mass storage devices.

`System_msus$` - STRING containing the msus of the mass storage device containing the system software. The value of this variable is set in the start-up routines when the system is loaded and run.

`Msus$` - STRING array of 5 string variables. Each string may contain the msus of a mass storage device connected to the system. System_msus$ = Msus$(1).

`N_msus` - INTEGER specifying the number of msus currently stored in Msus$.

The next set of variables define the EWE unit table. The EWE unit table is a collectgion of information which is needed by the EW environment. The primary use of the table is to tell the system where the EWE volumes are located. A maximum of 50 volumes are supported, 47 of which are used to store EWE files. The remaining 3 volumes are reserved for the system use:

```
#1   System Keyboard
#2   System console (CRT)
#6   System printer
```

The values of the unit table are created by the Set_volumes routine in the Start file.

`Ewe_select$` - STRING ARRAY consisting of 50 string variables, 10 characters each. The value of each string represents the select code and address of a peripheral, the msus of a mass storage device, or the name of the hardware volume. The values of this variable are set by the routine Set_volumes.

`Ewe_offset` - REAL ARRAY identifying the directory offset for the unit table. The value of each item in the array represents the starting record number of the volume. It is set to 0 for units #1, #2, and #6.

`Ewe_file_name$` - STRING ARRAY consisting of 50 string variables, 6 characters each. This is the actual BASIC file name of each EWE volume. This information is not required by the EW environment, but is used by EGS-45 for identification purposes.

`N_units` - INTEGER variable identifying the number of EWE volumes currently on line. The values of this variable range from 6 to 50.

Example Unit Table

Unit #	Ewe_select$	Ewe_offset	Ewe_file_name$
#1	KEYBOARD	0	
#2	CONSOLE	0	
#3	:F8,0	60	EWSYS
#4	:F8,1	60	EWPC
#5	:F8,1	1000	EWSCH
#6	7,1	0	PRINT
#7	:Q6,0	560	EWTOM
#8	:Q6,0	4560	EWTERY

N_units = 8

The next set of variables contain information about the control file.

Control_file$ - STRING ARRAY containing the name of control files. The control file identifies each module accessed by EGS-45.

Trees$ - STRING ARRAY of 12 strings. Each string variable contains the module name specified by the control file.

N_applications - INTEGER specifying the number of modules displayed in the Manager menu.

Branches$ - STRING ARRAY identifying the names of each branch available from each module.

Branches_name$ - STRING ARRAY identifying the BASIC program file name associated with each branch of a module.

N_branches - INTEGER ARRAY identifying the number of BASIC program files accessed by each module.

Iv1,Iv2 - Location of the current menu item in the Tree$ and Branches$ arrays.

Row,Col - Row and column of the current menu item.

Present_menu - This is the number of the present menu. If there is more than one set of menus needed, this variable identifies which menu is currently active.

Current_command$ - STRING variable containing the last command executed in the manager program. This variable is saved because it is necessary to highlight the command when returning to the manager.

Max_characters - INTEGER variable specifying the maximum number of characters per menu item. Currently set to 30.

Max_lines - INTEGER variable specifying the maximum number of lines per page for each menu. Currently set to 80.

Screen_width - INTEGER variable specifying the maximum number of characters that may be displayed in one line on the CRT.

Space - INTEGER variable specifying the the space between tree branches. Currently set to 5.

Manager$ - STRING variable containing the name of the manager program and the msus where it is located (for example, EGS45:F8). The manager program is stored on the same mass storage device as the system software.

Programming Tools

It is anticipated that many of the application programs that you write will need to read an EWE file created with the EGS-45 software. For example you may want to read the contents of a generate or archive file, post process the data found in the file, and output the processed data to an I/O device (such as a tape punch, a photo plotter, or an N/C drill press).

You could use the File Utilities module to transfer the contents of an EWE file into a BASIC file and then process the contents of the BASIC file in a BASIC language program. However, this requires loading and running the EGS-45 software, which is not always convenient or timely. As an alternative, HP has created a set of programming tools, PASSIGN and PREAD. These tools allow you to read an EWE file in the same fashion as you would read a BASIC data file with the BASIC language ASSIGN and READ statements.

PASSIGN and PREAD are BASIC language subprograms that are stored with the system software in the BASIC data files, PASSIG and PREAD. These subprograms must be added to your application program before they may be used. PASSIG may be appended to your BASIC language program by loading your program into the computer's memory and then typing:

```
GET "PASSIG",nnnn
```

where nnnn is the greater than the last line number of your application program. To addend PREAD to your application program, enter:

```
GET "PREAD",xxxx
```

where xxxx is greater than the new last line number of your application program. This causes the PASSIGN subprogram to be added to your program beginning at line number nnnn and the PREAD subprogram to be added to your program beginning at line number xxxx.

Before using the PASSIGN and PREAD subprograms, you should understand the use of the ASSIGN and READ statements to read BASIC data files in the BASIC programming language. You should also understand the use of subprograms in the BASIC language. This information is found in your computer's Operating and Programming manuals.

The PASSIGN Subprogram

Before an EWE file may be read by a BASIC program, you must assign the file with a call to the PASSIGN subprogram. This identifies the EWE file that you wish to read and provides the computer with the needed parameters to do so. The name of the EWE file to be read and the EWE volume that contains the file must be passed to the PASSIGN subprogram by placing these names in the variables `Pfile_name$` and `Efile_name$` before calling the PASSIGN subprogram. The syntax of the subprogram call and its variables are presented below.

```
CALL Passign (Pfile_name$,Efile_name$,Buf$,Npr,Lpr,Lrs,
File_type,Error)
```

The variables used by PASSIGN are described below.

`Pfile_name$` - This variable should contain the name of the EWE file to be read. No volume specifier is allowed with the file name. You must assign the EWE file name to the variable `Pfile_name$` **before** calling the subprogram, PASSIGN.

`Efile_name$` - This variable should contain the name of the BASIC file (EWE volume) that contains the EWE file to be read. A mass storage unit specifier is required with the file name (for example, `EWSYS:F8,1` or `EW1:H10` or `EWMINE:F8`). You must assign the BASIC file name (EWE volume name) to the variable `Efile_name$` **before** calling the subprogram, PASSIGN.

`Buf$` - The value of this variable is assigned by the PASSIGN subprogram and passed to the PREAD subprogram.

`Npr` - The value of this real variable is assigned by the PASSIGN subprogram and passed to the PREAD subprogram.

`Lpr` - The value of this real variable is assigned by the PASSIGN subprogram and passed to the PREAD subprogram.

`Lrs` - The value of this integer variable is assigned by the PASSIGN subprogram and passsed to the PREAD subprogram.

`File_type` - The value of this integer variable is assigned by the PASSIGN subprogram and passed to the PREAD subprogram.

`Error` - The value of this integer variable is passed back to you by the PASSIGN subprogram. It contains the error number of any error encountered by the PASSIGN subprogram. Examining the value of this variable immediately after each call to the PASSIGN subprogram tells you what error, if any, occurred.

The error number contained in the variable corresponds directly to the error codes of the computer and ROMs. Check your computer's Operating and Programming manuals to find the meanings of the error number returned in this variable. If the value of the variable `Error` is zero then no error occurred.

The PREAD Subprogram

The PREAD subprogram allows you to read the contents of an EWE file from a BASIC language program. The EWE file must first have been successfully assigned (no errors encountered) by a call to the PASSIGN subprogram.

Each call to the PREAD subprogram returns a string of characters which is a line of the EWE file. Successive calls to the PREAD subprogram allow you to read the entire EWE file by returning successive lines of data. The data read from the EWE file is returned in the variable `Data$` after each call to the PREAD subprogram. When all of the data has been read from the EWE file, the value of `Error` is negative one (−1), indicating that end of file has been found. The syntax of the subprogram call and its variables is discussed below.

```
CALL Pread (Npr,Lpr,Lrs,File_type,Buf$,Data$,Error)
```

The variables used by PREAD are described below.

`Npr` - The value of this real variable is passed to the PREAD subprogram by the PASSIGN subprogram.

`Lpr` - The value of this real variable is passed to the PREAD subprogram by the PASSIGN subprogram.

`Lrs` - The value of this integer variable is passed to the PREAD subprogram by the PASSIGN subprogram.

`File_type` - The value of this integer variable is passed to the PREAD subprogram by the PASSIGN subprogram.

`Buf$` - The value of this real variable is passed to the PREAD subprogram by the PASSIGN subprogram.

`Data$` - String variable (256 bytes) containing one line of data read from the EWE file. You may increase the size of `Data$` by dimensioning it larger in your variable declaration section.

`Error` - The value of this integer variable is passed back to you by the PREAD subprogram. It contains the error number of any error encountered by the PREAD subprogram. Examining the value of this variable immediately after each call to the subprogram PREAD tells you what error, if any, occurred.

The error number contained in `ERROR` corresponds directly to the error codes of the computer and ROMs. Refer to your computer's Operating and Programming manuals to find the meanings of the error number returned in this variable. If the value of `Error` is zero then no error occurred. If the value of `Error` is negative then all of the data in the EWE file has been read and an EOF (End Of File) has been encountered.

Using PASSIGN and PREAD

The PASSIGN and PREAD subprograms may be used in any BASIC language program to read EWE files. To help you understand the use of these subprograms, an example is provided at the end of this section. The example shows the use of the PASSIGN and PREAD subprograms in a BASIC language program to read the contents of an EWE file.

When the PASSIGN and PREAD subprograms are used, at least 3500 bytes of memory must be reserved, with an ICOM statement. Additionally, PASSIGN and PREAD require that the variables used to pass parameters be declared and/or dimensioned in your program. Finally, the subprograms require that two ASSEMBLY language programs, FDES and MSIO, be loaded into the system using the ILOAD statement before either subprogram may be called. These programs are stored on the mass storage media with your system software. The statements used to reserve common memory, declare and dimension variables, and load the FDES and MSIO ASSEMBLY programs are shown below. These statements must appear in any BASIC language program that calls the PASSIGN and PREAD subprograms.

```
10   ICOM 3500 (or 3500 PLUS any memory needed by other programs
requiring an ICOM statement)
20   INTEGER Error,Lrs,File_type
30   REAL Npr,Lpr
40   DIM Buf$[2560],Data$[256]
50   ILOAD "FDES:msus"  (where msus is the msus of the device where FDES is stored)
60   ILOAD "MSIO:msus"  (where msus is the msus of the device
where MSIO is stored)
```

Example

```
10  ICOM 3500     !This reserves 3500 bytes of memory for the subprograms
20  INTEGER Error,Lrs,File_type    !These program lines declare the variables
30  REAL Npr,Lpr                   !used by the subprograms to pass parameters
40  DIM Buf$[2560],Data$[256]      !and must present in any BASIC program
50  !                               using the PASSIGN and PREAD subprograms.
60  !
70   ILOAD "FDES:H10,1"   !Load the ASSEMBLY language programs required from
80  ILOAD "MSIO:H10,1"   !the mass storage device with msus = :H10,1
90  !
100 !
110 Efile_name$="EWSYS"   !Assign the name EWSYS as the EWE volume to be read
120 Pfile_name$ "Example" !Assign the name EXAMPLE as the EWE file to be read
130  !
140 !
150 CALL Passign(Pfile_name$,Efile_name$,Buf$,Npr,Lpr,Lrs,File_type,Error)
160 If Error<>0 THEN Error_routine !If the value of Error is anything besides
170                                 !zero, then goto Error_routine
180 !
190 Read_data:        !Loop to read all of the data in the EWE file, EXAMPLE.
200 Call Pread(Npr,Lpr,Lrs,File_type,Buf$,Data$,Error) !Call PREAD to read
210                                 !one line of the EWE file (up to 256 bytes)
220 !
230 If Error=-1 THEN Eof  !If the value of the variable Error equals -1, then
240                       !you have read all of the file.  Goto Eof routine.
250 !
260 IF Error <>0 THEN Error_routine
270 !
280 PRINT Data$       !This causes the line of the EWE file returned by the
290                   !subprogram PREAD, in the variable Data$, to be printed.
300 GOTO Read_data        !Loop to read again
310 !
320 Error_routine:      !
330 PRINT "ERROR";Error
340 STOP
350 !
360 Eof:     !
370 PRINT "*** END OF LIST ***"
380 END
390   !PASSIGN and PREAD should be appended to the end of the program by
400   !typing GET "PASSIG",1000 and GET "PREAD",5000 (this insures that
410   !PREAD is placed after the PASSIG routine).
```

Table of Contents

Chapter 7
The Schematic Drawing Module

Introduction

The Schematic Drawing Module is an application module for the Engineering Graphics System, EGS-45. It provides the tools to create and plot schematic drawings and generate material lists from these drawings. The module requires the same hardware set as the Engineering Graphics System and requires that the EGS-45 software be present on one of the system's mass storage devices.

Overview

With the Engineering Graphics System manual you learned to use the Graphics Editor to create drawings. You learned that you could customize the Graphics Editor for a specific application by creating menu, macro, and process files. You also learned that you could write your own BASIC language application programs that act as modules for the Engineering Graphics System. The Schematic Drawing Module is an example of these techniques.

While the module utilizes the EGS-45 Graphics Editor, the Schematic Drawing Module provides the menu, macro, and process files that customize the Graphics Editor for creating schematic drawings. A set of commonly used schematic symbols are provided as library parts with the module. A material list program provides a list of all components of a schematic drawing. This list may be sorted by either reference designator or by stock number.

With the Schematic Drawing Module you should have received:

- Schematic Drawing Module manual (HP part# 98302-12111)
- Schematic Drawing Module software cartridge (HP part# 98302-12504).

Please check to insure that you received all the items in the list above. If any items are missing, please contact your local HP Sales office.

This chapter of the manual assumes that you are familiar with the Graphics Editor and its subsytems. Additionally, it assumes that you understand the concepts pertaining to the EW environment and EWE files and volumes.

Preparing the Software

The Schematic Drawing Module software is supplied on a magnetic tape cartridge. Since this type of mass storage is not supported by the Engineering Graphics System, it is necessary to transfer the programs and files from the tape cartridge to a supported mass storage medium. The BASIC language program COPY is provided on the software cartridge to aid you in this transfer. To run the copy program, do the following:

1. Reset the computer to its initial state by either simultaneously pressing (CONTROL) (STOP) or by turning the computer off and then turning it on again.
2. Insert the Schematic Drawing Module software cartridge containing the Schematic Drawing Module software into the right hand tape drive of your computer.
3. Make sure that the disc in your mass storage device has been initialized. The disc may contain other files, but should have at least 200,000 bytes of storage space available.
4. Enter the following command on your computer:

   `LOAD "COPY:T15",1` (EXECUTE)

5. The computer next prompts you for the msus (Mass Storage Unit Specifier) of the source device. This is the mass storage device containing the software to be copied. You should respond by entering:

   `:T15` (CONTINUE)

6. Next you are prompted for the msus of the destination device. This is the mass storage device to which you wish to transfer the software. Enter the appropriate msus and press (CONTINUE).
7. Next the system displays the source and destination msus that you entered. It then prompts you to indicate if these msus are correct. Responding that the msus are not correct causes the system to again prompt you for the source and destination msus.
8. Press (C) to initiate the software transfer of all files on the source device to the destination device.*

Your disc now contains several BASIC files and the EWE volume EWSCH. EWSCH contains the macro, menu, process, and device files (library parts) that you will use to create schematic drawings. Once you have copied the Schematic Drawing Module software and files to a supported mass storage medium, you are ready to begin using the module to create schematics.

* Should the copy program encounter an existing file with the same name as the file being copied (as might occur when copying a new software release to the disc), you have three options.
1. Purge the existing file before making the copy or do not copy.
2. Automatically purge all existing files before performing the copy.
3. Skip duplicate files and continue with the next file.
The options are selected by pressing C, P or S, as instructed by the program.

Drawing Schematics

To create schematic drawings, you will need to load and run EGS-45. If the mass storage device containing EWSCH is part of the current system configuration, the volume EWSCH will automatically be recognized by the system. Otherwise you must select the System Configuration Module and reconfigure the system.

When creating schematics, you will want to select the Graphics Editor in the Schematic Drawing Module. This causes the supplied menu, macro, and process files to be loaded, thus customizing the Graphics Editor for the task. Selecting the Graphics Editor from any other module causes different process, menu, and macro files to be loaded. This changes the scale and spacing of the user grid, the layer definitions, the items listed in the menu, and available macros. Use of the Graphics Editor is examined in great detail in Chapter 2 of this manual. Therefore, discussion of the Graphics Editor in this chapter is limited to its applications in schematic drawing generation.

Schematic drawings are created by using the commands of the edit subsystem to add graphic components (such as lines, circles, and arcs) to the CRT. These components may be placed in the drawing so as to represent standard schematic symbols. Text and notes may be added to the drawing to provide textual information. The drawing may be plotted on any supported HP plotter by using the PLOT command in the edit subsystem.

A short example of creating a simple schematic is provided in the following section.

A Simple Example

In this example, a simple NAND gate will be created using the ADD command in the edit subsystem. Each individual command and the result of that command are detailed below.

1. Load and run the EGS-45 software.
2. Enter:

   ```
   GRID 0.05,5 0,0 ;
   GRID ;
   ```

   This sets grid points every .05 centimetres (since the process file for this module defines 100 internal units equal to one centimetre) and causes every fifth grid point to be displayed. The grid is aligned with the drawing origin 0,0.
3. Enter:

   ```
   ADD L1 1.25,0.25  0.50,0.25  0.50,-0.75  1.25,-0.75 ;
   ```

   This adds a line to the drawing which represents the input end of the NAND gate.
4. Enter:

   ```
   00,0 0.50,0 ;
   ```

   This adds a second line which represents an input line to the NAND gate.
5. Enter:

   ```
   00,-0.50 0.50,-0.50 ;
   ```

   This adds a third line which represents the second input of NAND gate.

6. Enter:

```
ADD A1 :R15 1.25,0.25 1.25,-0.75 1.75,-0.25 ;
```

This adds an arc which represents the output of the NAND gate.

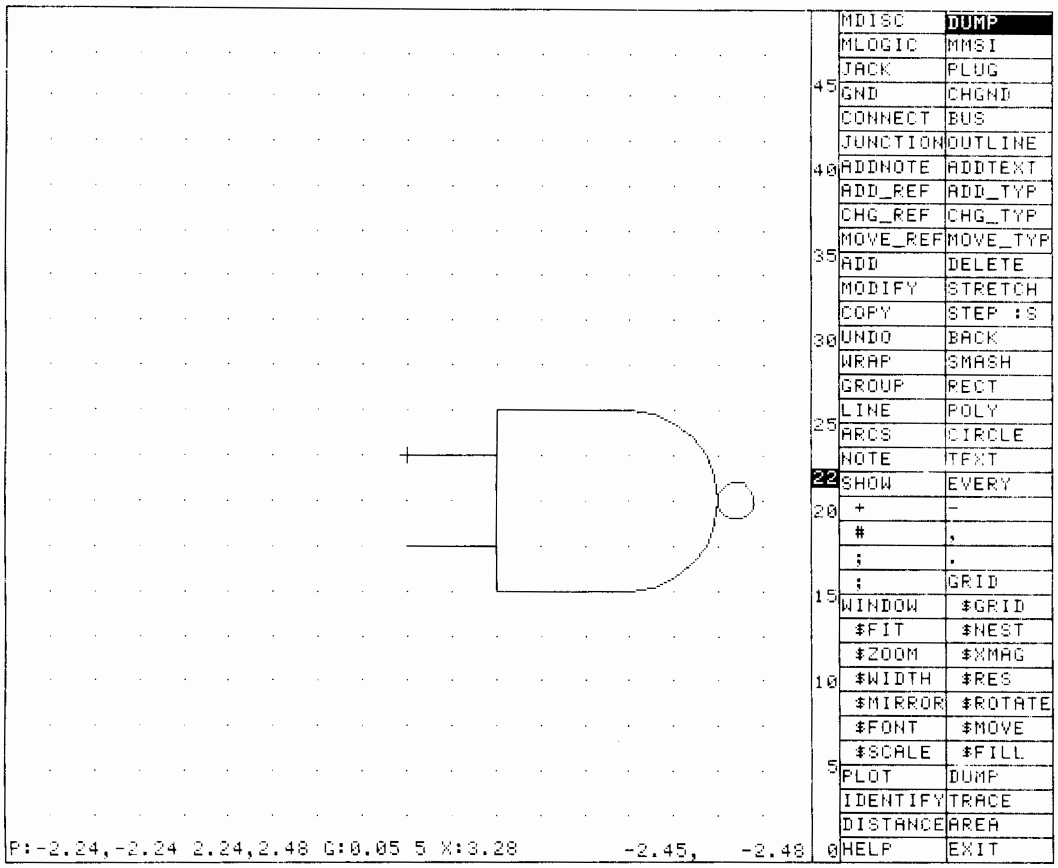
7. Enter:

```
ADD C1 :R15 1.85,-0.25 1.95,-0.25 ;
```

This adds a circle representing the inverter at the output of the NAND gate.

8. Obtain a printed copy of your drawing by entering:

```
DUMP
```



Similarly, by adding combinations of base level components to the drawing, any schematic symbol may be created. Using the WRAP, GROUP, and SAVE commands of the edit subsystem, you may create library parts (devices stored in EWE device files) that represent most standard schematic sumbols. You may then add instances of these devices to your drawing instead of simple graphic components, thus saving time and effort when creating a schematic.

Note the placement of the device origin in the example above. The device was constructed such that its origin is at the endpoint of one of the input lines. Remember that when an instance of a device is added to a drawing, it is added such that the origin of the linked device is placed at the xy location specified with the ADD command. This allows you to add the device at the exact location you desire. Thus, when adding an instance of the device created above, you would add the instance to the endpoint of a data line. This causes the origin of the device (the endpoint of an input line) to fall exactly at the endpoint of the data line.

Note

To use the material list program provided with this module, you must adhere to the drawing conventions imposed by this application program. Please review these conventions at the end of this section before attempting to lay out a schematic with the graphics editor.

Library Parts

To help you get started drawing schematics, a set of library parts is provided with this application module. These library parts are stored on the volume EWSCH and may be accessed in the same manner as any other device.

The supplied devices (library parts) consist of simple graphic components placed in layers 1 and 2. Components used in building the outline of the parts are placed in layer 1. Text and note components which make the library parts easier to understand are placed in layer 2. The choice of layers for device construction is arbitrary and is not dictated by the module. Layer use and definition is discussed in greater detail with the process file, SCH.PRO.

The library parts supplied with the module can be grouped into six general categories. The members of each group are listed in the table below. A list of the edit subsystem commands used to create each library part along with a drawing of each part is provided at the end of this chapter. You will find it beneficial to examine each library part on the CRT to determine the relative use of each part for your application.

Active Devices

Library Part Name	Description
PNP	PNP transistor
NPN	NPN transistor
FET	FET transistor
OPAMP	Operation amplifier
DIODE	Diode
ZDIODE	Zener diode
VSOURCE	Voltage source
ISOURCE	Current source
DBRIDGE	Diode bridge

Passive Devices

Library Part Name	Description
RES	Resistor
POT	Potentiometer (variable resistor)
CAP	Capacitor
PCAP	Polarized capacitor (electrolytic capacitor)
IND	Inductor (choke or coil)

Digital Devices

Library Part Name	Description
AND2	2 input AND gate
AND3	3 input AND gate
AND4	4 input AND gate
NAND2	2 input NAND gate
NAND3	3 input NAND gate
NAND4	4 input NAND gate
OR2	2 input OR gate
XOR2	2 input Exclusive OR gate
NOR2	2 input NOR gate
NOR3	3 input NOR gate
NOR4	4 input NOR gate
INV	Inverter
INVSCH	Schmitt Trigger Inverter
BUF	Buffer
BUFSCH	Schmitt Trigger Buffer
JKFFPC	JK Flip-Flop with Preset and Clear
DFFPC	D Flip-Flop with Preset and Clear
SRLATCH2	SR Latch (2 input latch type-SN74LS279)
SRLATCH3	SR Latch (3 input latch type-SN74LS279)

Complete Chips

Library Part Name	Description
INV240	Octal bus buffer/driver inverting (type-74LS240)
BUF244	Octal bus buffer/driver non-inverting (type-SN74LS244)
CNT290	4-bit binary and decade counter (type-SN74LS290)
CNT293	4-bit binary and decade counter (type-SN74LS293)
CNT190	4-bit binary and decade up/down synchronous counter (type-SN74LS190)
CNT193	4-bit binary and decade up/down synchronous counter (type-SN74LS193)
DCODE139	Decoder/Multiplexer (type-SN74LS139)
DSEL258	Quadruple 2-line-to-1-line Data Selector/Multiplexer (type-SN74LS258)

Drawing Enhancements

Library Part Name	Description
GND	Ground (system ground)
CHGND	Chassis ground
JUNCPT	Dot used to show an interconnection at the junction of two wires
JACK	Edge connector, in
PLUG	Edge connector, out
PAGECON	Page connector

System's Library Part

Library Part Name	Description
ID	Identification part (see section on system's library part below)

The library parts were created using a scale defined by the following UNITS command in the process subsystem:

```
UNITS  CM,100 ;
```

This sets 100 system grid points equal to one centimetre. The user and display grids with which the library parts were created are defined by:

```
GRID .05,5  ;
```

This specifies that a user grid point exists every .05 centimetres on the system grid. Every 5th user grid point is displayed.

Each device may be used on a drawing with any grid and scale. However, the recommended setting for adding instances of library parts and for adding the connection lines between these devices is defined by:

```
GRID .25,1
```

The above grid statement causes the endpoints of the input and output lines of the supplied devices to lie on user grid points, when added. This makes it easier to add connection lines between these devices and the other components forming your drawing.

The above information should be helpful when modifying a supplied device (library part). Additionally, you will find this information useful when attempting to add an instance of a supplied device to a drawing with a different units definition or a different grid alignment. You may also wish to change the library part origin, since it is the origin of the device that is placed at the xy location point specified with the ADD command.

Existing library parts may be modified with the following procedures:

1. Enter the edit subsystem by entering the EDIT command along with the name of the library part name. For example:
   ```
   EDIT NAND2 ;
   ```
2. Make any changes to the device by using the edit subsystem commands.

   To change the origin of a library part, use the WRAP command to wrap the entire library part, thus creating an unnamed device. Next use the MOVE command to position the library part on the drawing such that the origin you desire for the library part is coincidental with the drawing origin, 0,0.
3. Save the library part using the SAVE command.

CAUTION

MODIFYING A LIBRARY PART WILL CAUSE **EVERY** INSTANCE OF THAT DEVICE IN EVERY DRAWING UTILIZING THAT DEVICE TO REFLECT THAT CHANGE.

System's Library Part

A system's library part is a device whose sole use is to relay information from the drawing to the material list. For example, this information may consist of the drawing number, the designer's name, or the drawing creation date. The information is assigned as the iname and/or the tname of the system's library part.

The supplied system's library part is a device named ID which consists of a single note component. The note component consists of the characters `ID-`. You may modify this device to change the font size of the note and characters that form the note. Additionally, you may create your own system's library part. Any device named ID serves as a system's library part. The system reads the iname and tname of the system's library part and places this information in the allocated space in the material list report.

For example, suppose that you add an instance of a library part named ID to your drawing and then create a material list from that drawing. Assigning 0123-4567 (the drawing number) and October 8, 1981 (the drawing date), as the iname and the tname of that part causes the heading of the material list to be:

```
                                Material List
                             <archive file name>


0123-4567
October 8, 1981


<one line header>
                                                             <today's date>
```

If more than one instance of the system's library part occurs in your drawing, the system takes the information from the last system's library part to occur in the archive file.

If the system's library part is not used in your schematic, no space is allocated for it in the material list header.

The Process File

The process file SCH.PRO, is supplied with the Schematic Drawing module on the volume EWSCH. This file shows how you might define the layers to partition the information contained in a schematic drawing. Partitioning information in your drawing allows you to view certain parts of your drawing without having to view the entire drawing. By turning layers on and off with the SHOW command you can limit the amount of information displayed on the drawing and thus decrease the amount of time needed to re-draw the screen. Additionally, turning on and off the display of components within a layer allows you to view selected parts of your drawing, thus saving re-draw time.

For example, suppose that all of the graphic components (the lines, arcs, etc) that form the drawing are placed in layer 1. And suppose that all text or note components (such as pin numbers) are placed in layer 2. By entering `SHOW -E2 ;`, no component in layer 2 is displayed. If other components share layer 2 with the text and/or note components, entering `SHOW -T2,-N2 ;` causes all components except text and note components to be displayed.

When the Graphics Editor is selected from the Schematic Drawing module, the process file SCH.PRO is automatically loaded. If you wish to have your own customized process file automatically loaded with the graphics editor, you should name it SCH. Alternately, you may modify the control file for this module (see Chapter 6 for details) to change the name of the process file that is automatically loaded. To modify the BASIC environment control file, enter:

```
GET CHR$(1)&"SCH:msus" (EXECUTE)
```

where msus is the mass storage unit specifier of the device containing the Schematic Drawing module software. Modify the control file and re-save it by entering:

```
RE-SAVE CHR$(1)&"SCH:msus" (EXECUTE)
```

A listing of the supplied process file SCH.PRO is provided below:

LAYER #	PEN #	LABEL	COLOR	LINE TYPE	LAYER TYPE	INAME Disp Size	TNAME Disp Size
0	1	INSTANCE	White	Solid	Boundary	ON 0.30	ON 0.30
1	1	PARTS	Red	Solid	Detail	OFF 1.00	OFF 1.00
2	2	NOTES	Yellow	Solid	Detail	OFF 1.00	OFF 1.00
3	3	CONNECTS	Green	Solid	Detail	OFF 1.00	OFF 1.00
4	4	BUSLINE	Aqua	Solid	Detail	OFF 1.00	OFF 1.00
5	5	JUNCT	Yellow	Solid	Detail	OFF 1.00	OFF 1.00
6	6	NOTES	Purple	Solid	Detail	OFF 1.00	OFF 1.00
7	7	OUTLN	Yellow	Broken	Detail	OFF 1.00	OFF 1.00

The Macro File

The macro file SCH.MAC is supplied with the Schematic Drawing module on the volume EWSCH. Some of the macros are designed to aid the user who intends to use the material list program. They provide a convenient single command to assign, change, and place the iname and tname as required by the material list progam. Other macros are provided to simplify the more cryptic parameters with functional one word macros. For example, the macro `$ZOOM` replaces the parameter `:Z` with the `WINDOW` command. The function of `$ZOOM` is more intuitively obvious than that of `:Z`. Other macros load new menu files for the edit subsystem. The menu files contain the names of the supplied library parts. This allows you to select the name of the library part instead of typing it when adding instances of a device.

When the Graphics Editor is selected from the Schematic Drawing module, the macro file SCH.MAC is automatically loaded. If you wish to have your own customized macro file automatically loaded with the graphics editor, you should name it SCH. Alternately, you may modify the control file for this module (see Chapter 6 for details) to change the name of the macro file that is automatically loaded. To modify the control file, enter:

`GET CHR$(1)&"SCH:msus"` (EXECUTE)

where msus is the mass storage unit specifier of the device containing the Schematic Drawing module software. After modifying the control file, re-save the file by entering:

`RE-SAVE CHR$(1)&"SCH:msus"` (EXECUTE)

Since the definition of each macro may be obtained by listing the contents of the macro file, only the general use and purpose of each macro is discussed here. Functionally, the macros may be divided into three groups.

One group of macros provided in the macro file is intended to simplify the use of parameters and descriptors with commands in the edit subsystem. For example, the function of `WINDOW :G ;` is not obvious from the command syntax. However, if a macro $GRID is created for the :G parameter, the function of the command becomes much more obvious (`WINDOW $GRID ;`). Many macros of this type are provided and each is described below.

ARCS - macro which replaces the arc component descriptor A. For example, `ADD ARCS 1 0,0 50,0 10,10 ;` or `SHOW -ARCS 5;`.

BACK - macro which replaces the backup command, >. For example, to delete an xy point entered with the ADD command, enter `ADD ARCS 1 0,0 100,100 20,20 BACK ;`.

CIRCLE - macro which replaces the circle component descriptor C. For example, `ADD CIRCLE 1 0,0 50,50 ;` or `MOVE CIRCLE 0,0 20,20 75,75 ;`.

EVERY - macro which replaces the descriptor for every component E. For example, `SHOW #EVERY ;`.

$FILL - macro which replaces the parameter :F (fill option) with the PLOT command. For example, `PLOT $FILL ;` .

$FIT - macro which replaces the :F parameter (fit option) with the WINDOW command. For example, `WINDOW $FIT ;`.

$FLIPY - macro which replaces the :MY parameter (mirror about Y axis). This macro is used with any command where :MY is a legal option. For example, `ADD EXAMP1 $FLIPY 0,0 ;`.

$FONT - macro which replaces the :F parameter (font size option) with all edit subsystem commands which allow you to specify a font size. For example, `ADD T1 $FONT 15 "TEXT" 0,0 ;`.

$GRID - macro which replaces the :G (grid display option) with the WINDOW command. For example, `WINDOW $GRID ;`.

INSTNCE - macro which replaces the instance component descriptor I. For example, `MOVE INSTNCE EXAMPL 0,0 0,0 100,100 ;`.

LINE - macro which replaces the line component descriptor L. For example, `ADD LINE 1 0,0 20,20 50,50 ;` .

$MIRROR - macro which replaces the :M parameter (mirror option) with all edit subsystem commands which allow this option. For example, `ADD T1 $MIRRORX "TEXT" 0,0 ;` (mirrors the text about the X axis).

$MOVE - macro which replaces the :M parameter when moving the location of inames and tnames. For example, `TRACE $MOVE 0,0 50,50 ;`.

$NEST - macro which replaces the :N parameter (nesting depth display option) with the WINDOW command. For example, `WINDOW $NEST3;` .

NOTE - macro which replaces the note component descriptor N. For example, `ADD NOTE 3 $FONT10 "WORDS" 0,0 ;`.

POLY - macro which replaces the polygon component descriptor P. For example, `ADD POLY 1 0,0 50,50 80,30 ;`.

RECT - macro which replaces the rectangle component descriptor R. For example, `ADD RECT 1 0,0 100,100 ;` or `DELETE RECT 25,25 ;`.

$RES - macro which replaces the :R parameter used to specify resolution of circle and arc components, when added. For example, `ADD C2 $RES 6 0,0 50,50 ;`.

$ROTATE - macro which replaces the :R parameter (rotate option) with all edit subsystem commands which allow rotation. For example, `ADD EXAMPL $ROTATE 45 0,0 ;` specifies to add an instance of the device EXAMPL at the location 0,0 and rotate the device 45 degrees about its origin.

$ROTCW - macro which specifies to rotate the component 90 degrees clockwise. This macro is valid with all edit subsystem commands which allow rotation. It is equivalent to entering the parameter/data combination :R -90. For example, `ADD T1 $ROTCW "TEST" 0,0 ;` adds the text TEST at the location 0,0 and rotates it 90 degrees clockwise about its origin.

$ROTCCW - macro which specifies to rotate the component 90 degrees counter-clockwise. This macro is valid with all edit subsystem commands which allow rotation. It is equivalent to entering the parameter/data combination :R 90. For example, `ADD T1 $ROTCCW "TEST" 0,0 ;` adds the text TEST at the location 0,0 and rotates it 90 degrees counter-clockwise about its origin.

$SCALE - macro which replaces the :S parameter (scale option) with the ADD command (for adding instance components) and the PLOT command. For example, `ADD EXAMPL $SCALE 2 0,0 ;`.

TEXT - macro which replaces the text component descriptor T. For example, `ADD TEXT 1 "TEST" 0,0 ;`.

$XMAG - macro which replaces the :X parameter (absolute magnification option) with the WINDOW command. For example, `WINDOW $XMAG15 ;` .

$WIDTH - macro which replaces the :W parameter (width option) when adding lines with the ADD command. For example, `ADD LINE 2 $WIDTH 20 0,0 50,50 ;`.

$ZOOM - macro which replaces the :Z parameter (zoom option) with the WINDOW command. For example, `WINDOW $ZOOM 3;` .

Another group of macros provides the ability to load any one of four supplied menus for the edit subsystem. This technique, called menu paging, allows you to easily load a different set of commands, parameters, and device names in the menu slots. This allows these items to be selected from the menu. Each menu is described and shown in the menu file section of this chapter. The macros, their definitions and their functions are shown below.

Macro	**Equivalent Edit Subsystem Command(s)**	**Function**
MMSI	`MENU EWSCH:MSI;`	Loads the menu file, MSI.MEN, from the volume, EWSCH.
MLOGIC	`MENU EWSCH:LOGIC;`	Loads the menu file, LOGIC.MEN, from the volume, EWSCH.
MDISC	`MENU EWSCH:DISCRETE;`	Loads the menu file, DISCRETE.MEN, from the volume, EWSCH.
HOME	`MENU EWSCH:SCH;`	Loads the menu file, SCH.MEN, from the volume, EWSCH.

A third group of macros provides capabilities for assigning, changing, and moving the tname and iname associated with a graphic component (such as a line or an instance of a library part). Additionally, a few macros have been created that show how macros can be used to speed drawing entry. These macros, their definitions, and their functions are described below.

ADDNOTE - Macro which adds a .2 user unit high note component to layer 6 of the drawing. To use this macro, enter `ADDNOTE` followed by the characters forming the note component. These characters must be enclosed by quotation marks. Next select the xy location where the note component is to be placed. Entering additional xy locations allows you to change the location of the note component. For example,

```
ADDNOTE "ABC" 0,0 ;
```

adds the note ABC to the location 0,0.

ADDTEXT - Macro which adds a .2 user unit high text component to layer 6 of the drawing. To use this macro, enter ADDTEXT followed by the characters forming the text component. These characters must be enclosed by quotation marks. Next select the xy location where the text component is to be placed. Entering additional xy locations allows you to change the location of the text component. For example,

```
ADDTEXT "ABC" 0,0 ;
```

adds the text ABC to the location 0,0.

Note

The CHG_REF, CHG_TYP, ADD_REF, and ADD_TYP macros already have the quotes required to enter text in the macro. This was done for convenience. This feature, however, has one limitation. If the REF or TYP has any spaces in it, there will be an error. The system always assumes that any spaces in a parameter list in a macro are delimiters. To enter REF and TYP with spaces imbedded, do not use macros, but instead use the IDENTIFY and TRACE command directly (refer to IDENTIFY and TRACE commands for correct syntax).

ADD_REF - Macro which allows you to add an iname to a component. The iname is used by the material list program to specify the reference designator of the electrical part. To use this macro, enter ADD_REF followed by the characters of the reference designator (without quotation marks). Next select the an xy point that lies in or on the component to which the reference designator is to be assigned. If the point selected does not lie in or on the component an error message to that effect is displayed and the macro must be re-entered.

The display of the designator occurs at the location specified when selecting the component. Entering additional xy locations that lie in or on the component changes the location for the display of the designator (acts as a trial placement point). For example,

```
ADD_REF U1 0,0 ;
```

assigns the reference designator U1 to the component located at 0,0. The designator is placed at the location 0,0 when displayed.

ADD_TYP - Macro which allows you to add a tname to a component. The tname is used by the material list program to specify the electrical part type. To use this macro, enter ADD_TYP followed by the characters of the electrical part type (without quotation marks). Next select the an xy point that lies in or on the component to which the part type is to be assigned. If the point selected does not lie in or on the component an error message to that effect is displayed and the macro must be re-entered.

The display of the part type occurs at the location specified when selecting the component. Entering additional xy locations that lie in or on the component changes the location for the display of the part type (acts as a trial placement point). For example,

```
ADD_TYP SN74LS09 0,0 ;
```

assigns the electrical part type SN74LS09 to the component located at 0,0. The part type is placed at the location 0,0 when displayed.

BUS - Adds a line with a width of .04 user units to layer 4. This line represents a bus on a schematic, according to the conventions recommended with the material list program. To use this macro, enter BUS followed by the xy locations for the first endpoint of the bus, any vertices of the bus line, and finally, the second endpoint of the bus. For example,

```
BUS 0,0 50,50 0,100 ;
```

adds a .04 user unit wide line to layer 4 of the drawing. The bus line starts at 0,0, has a vertex at 50,50, and ends at 0,100.

CHG_REF - Replaces the existing iname of a component with the iname supplied. The iname is used by the material list program to specify the reference designator of an electrical part. To change the reference designator of a component, enter CHG_REF and the new reference designator (do not enclose the designator in quotation marks). Next, select an xy location that lies in or on the component whose reference designator you wish to change. The iname of the component is changed to the new iname supplied. For example,

```
CHG_REF NEW 0,0
```

changes the iname of the component located at 0,0 to NEW.

CHG_TYP - Replaces the existing tname of a component with the tname supplied. The tname is used by the material list program to specify the electrical part type. To change the part type of a component, enter CHG_TYP and the new electrical part type (do not enclose the part type in quotation marks). Next, select an xy location that lies in or on the component whose part type you wish to change. The tname of the component is changed to the new iname supplied. For example,

```
CHG_TYP NEW 0,0
```

changes the tname of the component located at 0,0 to NEW.

CONNECT - Adds a line to layer 3 which represents a connection between parts on a schematic drawing. Layer 3 is used to be consistent with the process file, SCH.PRO. To use this macro, enter CONNECT and then select one endpoint of the connection line. Next, select any vertices for the line. Finally, select the second endpoint of the connection line. For example,

```
CONNECT 0,0 50,0 50,50 ;
```

adds a line to layer three that has 0,0 as an endpoint, 50,0 as a vertex, and 50,50 as the final endpoint.

JUNCTION - Allows you to add an instance of the library part JUNCPT to your drawing. JUNCPT is used to represent the electrical connection of two wires. To use this macro enter JUNCTION and select the xy location where the instance is to be placed. For example,

```
JUNCTION 0,0 ;
```

adds an instance of JUNCPT at the location 0,0.

MOV_REF - Allows you to change the xy location where the iname of a component is displayed. To use this macro, enter MOV_REF and an xy location that lies in or on the component. Next enter the new location where the iname is to be displayed. For example,

```
MOV_REF 0,0  20,20 ;
```

changes the location of the iname of the component located at 0,0 to 20,20.

MOV_TYP - Allows you to change the xy location where the tname of a component is displayed. To use this macro, enter MOV_TYP and an xy location that lies in or on the component. Next enter the new location where the tname is to be displayed. For example,

```
MOV_TYP 0,0 20,20 ;
```

changes the location of the tname of the component located at 0,0 to 20,20.

OUTLINE- Adds a line to layer 7 in your drawing. Often schematics uses dashed lines to group parts of the drawing that are logically associated. To use this macro, enter OUTLINE and select the endpoints and vertices that form the line. For example,

```
OUTLINE 0,0 0,50 50,50 ;
```

adds a line to layer 7 of your drawing. The lines have endpoints of 0,0 and 50,50 with a vertex located at 0,50.

The Menu File

Four menu files are provided with the Schematic Drawing Module. These menu files customize the graphics editor for schematic drawing. Each menu holds a number of fundamental edit subsystem commands, macro names, and library part names. Note that a macro located in the menu may not be selected unless the macro file defining that macro is first loaded into the system.

The menu file SCH.MEN is automatically loaded when the Graphics Editor is selected from the Schematic Drawing module. If you wish to have your own customized menu file automatically loaded with the Graphics Editor, you should name it SCH. Alternately, you may modify the control file for this module (see Chapter 6 for details). This allows you to specify the name of the menu file which is automatically loaded. To edit the control file, enter:

```
GET CHR$(1)&"SCH:msus" (EXECUTE)
```

where msus is the mass storage identifier of the device containing the Schematic Drawing module software. After modifying the control file, re-save it by entering:

```
RE-SAVE CHR$(1)&"SCH:msus" (EXECUTE)
```

	HOME	DUMP
	MLOGIC	MMSI
	JACK	PLUG
45	GND	CHGND
	CONNECT	BUS
	JUNCTION	OUTLINE
40	ADDNOTE	ADDTEXT
	ADD_REF	ADD_TYP
	CHG_REF	CHG_TYP
	MOVE_REF	MOVE_TYP
35	ADD	DELETE
	MODIFY	STRETCH
	COPY	STEP :S
30	UNDO	BACK
	WRAP	SMASH
	GROUP	
	RES	POT
25	CAP	PCAP
	IND	
22	DIODE	ZDIODE
20	DBRIDGE	
	PNP	NPN
	FET	
15	OPAMP	
	VSOURCE	ISOURCE
		-
10	;	,
	:	.
	WINDOW	GRID
5	$FIT	$GRID
	$ZOOM	$NEST
	$XMAG	
	$ROTCW	$ROTCCW
0	DISTANCE	$FLIPY

MDISC

	MDISC	DUMP
48	HOME	MMSI
	JACK	PLUG
45	GND	CHGND
	CONNECT	BUS
	JUNCTION	OUTLINE
40	ADDNOTE	ADDTEXT
	ADD_REF	ADD_TYP
	CHG_REF	CHG_TYP
	MOVE_REF	MOVE_TYP
35	ADD	DELETE
	MODIFY	STRETCH
	COPY	STEP :S
30	UNDO	BACK
	WRAP	SMASH
	GROUP	
	NAND2	NOR2
25	NAND3	NOR3
	NAND4	NOR4
	AND2	OR2
20	AND3	XOR2
	AND4	
	INV	INVSCH
	BUF	BUFSCH
15	JKFFPC	DFFPC
	SRLATCH2	
	SRLATCH3	-
10	;	,
	:	.
	WINDOW	GRID
5	$FIT	$GRID
	$ZOOM	$NEST
	$XMAG	
	$ROTCW	$ROTCCW
0	DISTANCE	$FLIPY

MLOGIC

	MDISC	DUMP
48	MLOGIC	HOME
	JACK	PLUG
45	GND	CHGND
	CONNECT	BUS
	JUNCTION	OUTLINE
40	ADDNOTE	ADDTEXT
	ADD_REF	ADD_TYP
	CHG_REF	CHG_TYP
	MOVE_REF	MOVE_TYP
35	ADD	DELETE
	MODIFY	STRETCH
	COPY	STEP :S
30	UNDO	BACK
	WRAP	SMASH
	GROUP	
	DCODE139	DSEL258
25	CNT190	CNT193
	CNT290	CNT293
	INV240	BUF244
20		
15		
		-
10	;	,
	:	.
	WINDOW	GRID
5	$FIT	$GRID
	$ZOOM	$NEST
	$XMAG	
	$ROTCW	$ROTCCW
0	DISTANCE	$FLIPY

MMSI

	MDISC	DUMP
	MLOGIC	MMSI
	JACK	PLUG
45	GND	CHGND
	CONNECT	BUS
	JUNCTION	OUTLINE
40	ADDNOTE	ADDTEXT
	ADD_REF	ADD_TYP
	CHG_REF	CHG_TYP
	MOVE_REF	MOVE_TYP
35	ADD	DELETE
	MODIFY	STRETCH
	COPY	STEP :S
30	UNDO	BACK
	WRAP	SMASH
	GROUP	RECT
	LINE	POLY
25	ARCS	CIRCLE
	NOTE	TEXT
22	SHOW	EVERY
20	+	-
	#	,
	:	.
	;	GRID
15	WINDOW	$GRID
	$FIT	$NEST
	$ZOOM	$XMAG
10	$WIDTH	$RES
	$MIRROR	$ROTATE
	$FONT	$MOVE
5	$SCALE	$FILL
	PLOT	DUMP
	IDENTIFY	TRACE
	DISTANCE	AREA
0	HELP	EXIT

SCH

Conventions

So far, you have learned how to use the commands of the graphics editor to create a schematic drawing. If you only wish to draw schematics, this is all you need to know. However, if you wish to use the material list program to obtain a listing of all of the reference designators used on a schematic drawing or a list of all electrical parts required to build the circuit represented in the schematic, it is necessary to adhere to a set of conventions when creating the schematic. This set of conventions specifies a standard way of constructing schematics and assigning reference designators to the symbols representing the electrical part. These conventions are described below.

- Electrical Parts- Electrical parts (such as resistors and capacitors) must be represented by instances of library parts. The material list program recognizes only instances of library parts as representing electrical parts.
- Reference Designators- A reference designator is a combination of letters and numbers used to identify an electrical part on a schematic drawing. Typically, the letters (called a class designator) identify the type of electrical part represented. For example, R specifies a resistor, C specifies a capacitor, and L specifies an inductor (also called a coil or a choke). The numbers (called a unit number) identify the differences between electrical parts of the same type: R1, R2, R3, C1, C2, C3. Trailing letters (such as B in the designator U1B) are used to specify an individual component of a packaged device. For example, SN74LS09 is an IC package containing four 2-input AND gates. The reference designators U1A, U1B, U1C, and U1D might be used as reference designators for each individual AND gate.

Reference designators are assigned as the iname of an instance of the library part which represents the electrical part. The iname is the textual string associated with a graphic component. It is assigned by means of the ADD command or the IDENTIFY command. For example, `ADD NAND .U1A 0,0 ;` causes an instance of the library part NAND to be added to the drawing at the location 0,0. The iname (reference designator) U1A is assigned to the instance.

Additionally, the iname may be assigned, moved, or changed with the IDENTIFY command. The macros ADD_REF, MOV_REF, and CHG_REF are provided to make it easier to assign, change, and move the reference designator. Layer 0 in the supplied process file (the layer the system reserves for displaying instance components) is defined such that the iname is always displayed.

Reference designators may be up to 16 characters in length and may consist of the characters A thru Z, the integers 0 thru 9, and the characters: (*), ($), (?), (=), and (_). If the optional quotation marks are used when assigning the iname (see the ADD command or the IDENTIFY command in the syntax reference) additional characters may be used. Keep in mind that only a limited number of these additional characters may be plotted (see the PLOT command in the edit subsystem).

- Electrical Part Type- The electrical part type is a set of characters used to identify a physical electric part. We recommend the manufacturer's product number or some other device specific information be used as the electrical part type. For example, the electrical part type for a four input NAND gate might be SN74LS09 (the manufacturer's product number). Similarly, a 10K ohm resistor might use an electrical part type of 10K. Notice that a schematic drawing using five 10K resistors requires five different reference designators (R1, R2, ...) while only one electrical part type, 10K is needed.

  Electrical part types are assigned as the tname of an instance of a library parts which represents the electrical part. The tname is the textual string associated with the component. It is assigned by means of the ADD command or the TRACE command. For example, `ADD NAND @SN74LS09 0,0` causes an instance of NAND to be added to the drawing at the location 0,0. The tname (electrical part type), SN74LS09 is assigned to the instance.

  Additionally, the tname may be assigned, moved, or changed with the TRACE command. The macros ADD_TYP, MOV_TYP, and CHG_TYP are provided to make it easier to assign, change, and move the electrical part type. Layer 0 in the supplied process file (the layer the system reserves for the displaying instance components) is defined such that the tname is always displayed.

  Reference designators may be up to 16 characters in length and may consist of the characters A thru Z, the integers 0 thru 9, and the characters: (*), ($), (?), (=) and (_). If the optional quotation marks are used when assigning the iname (see the ADD command or the IDENTIFY command in the syntax reference) additional characters may be used. Keep in mind that only a limited number of these additional characters may be plotted (see the PLOT command in the edit subsystem).

An electrical part type may be up to 16 characters in length and may consist of the characters A thru Z, the integers 0 thru 9, and the characters: (*), ($), (?), (=) and (_). If the optional quotation marks are used when assigning the tname (see the ADD command or the TRACE command in the syntax reference) additional characters may be used. Keep in mind that only a limited number of these additional characters may be plotted (see the PLOT command in the edit subsystem).

- Bus Lines- Bus lines appear very often in schematics detailing memory board design. To eliminate the tedium of drawing a line for each line in the bus, the convention is defined that any line with non-zero width in a drawing is assumed to be a bus. Since the material list program does not consider buses to be electrical parts (and thus not considered for inclusion in the material list), the use of this convention is a matter of convenience. Representing buses with individual lines for each line in the bus will generate a material list identical to one generated from a drawing where a bus is represented according to the convention.
- Archive file- The material list program can only produce a material list from an archive file. This means that an archive file must be created from the device file containing your schematic drawing. Archive files are created with the ARCHIVE command in the Graphics Editor. For example, if your schematic drawing is saved in the device file EXAMPL.DEV, the following command creates the required archive file:

```
ARCHIVE EXAMPL ;
```

It is important that you NOT limit the nesting level depth when creating the archive file. The nesting levels of the library parts are essential to the material list program.

The Parts File

Although you now have a standard way to construct your schematic drawing and a standard way of assigning reference designators and electrical part types, this information alone is still not enough to generate a proper material list. The additional information needed to generate the material list is stored in **one or more** parts files. These parts files are constructed, modified, and maintained through the Edit Parts File branch of the Schematic Drawing module. A parts file is a BASIC data file. The contents of a parts file resembles a five column table. Each column (called a field) contains specific information about each library part entered in the parts file. Two sample parts files are supplied on the volume EWSCH: GPARTS (for Generic parts) and SPART (for Specific parts).

Each parts file contains the following information about each library part used in a drawing:

- Device name - This is the name of the library part representing an electrical part in your schematic drawing. Several different electrical parts may be represented with only one library part.

  For example, the library part RES (a resistor) may be used to represent a 10K resistor, a 50K resistor, and a 100 ohm resistor. The electrical part type that you assign to the instances of that part differentiates one electrical part from another (in the drawing, in the material list generated, and in the parts file). Thus your part file would need three separate entries for the device name, RES. The electrical part type for each entry would be 10K, 30K, and 100, respectively.

- Electrical part type - The information contained in this field specifies a particular electrical device (as opposed to the device name which specifies a generic electrical device). For example, suppose a schematic is created with two different sizes of resistors, 50K ohms and 100K ohms. Both resistors can be represented with an instance of the supplied library part RES, a generic resistor. By assigning an electrical part types of 50K and 100K to the appropriate resistors, you assign a physical characteristic to the generic device.

  The electrical part type that you enter in this field must match the electrical part type that you assign to the instance of the library part in your drawing. A maximum of 16 characters may be used for the electrical part type. It may be composed of the following characters: the letters A through Z, the numerals 0 through 9, (_), (*), (=), ($), and (?). Additionally, any of the following characters may be used: (<) (>) (%) (\) (!) (,) (.) (;) (+) (−) (#) (∧) (@) (&) (/) ([) (]) ( ). Although these special characters can be assigned, care should be exercised in selecting the characters for the part type since only a limited number of these can be plotted with the PLOT command (see the PLOT command for details).

- List Option - The information contained in this field specifies whether the entry is used when generating the material list. It consists of a single letter: Y (for yes, include in the listing) or N (for no, do not include in the listing). An entry in the parts file is required for each instance of a library part that occurs in your drawing. However, not every library part that appears on your drawing necessarily represents an electrical part. The list option field allows you to specify which devices are to be included in the material list generated.

  For example, a schematic drawing might use several instances of the library part, CHGND (chassis ground), which does not represent a physical electrical part. If no entry is made in the parts file for the device CHGND, all instances of that device are listed with an error message in the material list. However, if an entry for the device exists in the parts file and the list option is N for that entry, the device in not included in the material list at all.

- Stock number - The information in this field represents your company's stock number for the electrical part. For example, a schematic drawing contains an instance of the library part RES with the part type, 10K. This specifies a specific resistor. These resistors are stocked in the company stock room in bin number 05253-10562. Therefore the entry in the parts file for 10K resistors would consist of the following: `RES` (device name), `10K` (the electrical part type), `Y` (the list option), and `05253-10562` (your stock number for the part).

  The size of the stock number field varies from 8 characters to 20 characters and is defined at the time the parts file is created. The width of the stock number field in the supplied parts files is 11 characters. Any character appearing on your computer's typewriter keyboard or numeric pad may be used in the stock number field.

- Description - The information in this field is a description of the electrical part. This information may consist of a single word, a phrase, or a complete sentence. The description contained in this field is printed out with each electrical part in the material list. This helps the reader of the material list determine exactly which electrical part is specified by the device name, part type, and stock number.

  The size of the description field is equal to (51 - X) where X is the size of the stock number field. Since you specify the width of the stock number field when you create the parts file, you also specify the size of the description field. Any character appearing on your computer's keyboard or numeric pad may be used in the description field. The size of the description fields in the supplied parts files is 40 characters.

A portion of the supplied parts file SPART is provided below as an example. To see a complete listing of either supplied parts file, select Edit Parts File branch from the Manager menu. Next select GPART or SPART (the names of the parts files). This causes yet another set of options to be displayed. Select the LIST option to view the entire parts file.

```
Device                   List Stock     **** Parts file - SPART:F8,1 ****
 Name     Type              ? Number                 Description

AND2      74LS08            Y 1820-1201   74LS08 quad 2-input AND gate
AND3      74LS11            Y 1820-1203   74LS11 tpl 3-input AND gate
AND4      74LS21            Y 1820-1205   74LS21 dual 4-input AND gate
BUF       74LS368           Y 1820-1492   74LS368 hex 3-state inverting BUFFER
BUF244    74LS244           Y 1820-2024   74LS244 Octal non-inv bus buffer/driver
CNT190    74LS190           Y 1820-1279   74LS190 decade sync up-down COUNTER
CNT193    74LS7193          Y 1820-1194   74LS193 4-bit bnry sync up-down COUNTER
CNT290    74LS290           Y 1820-1442   74LS290 decade async COUNTER
CNT293    74LS293           Y 1820-1443   74LS293 4-bit binary async COUNTER
DCODE139  74LS139           Y 1820-1281   74LS139 2-to-4 line DEMUXR
DFFPC     74LS74            Y 1820-1112   74LS74 dual D-FF with Preset and Clear
DSEL258   74LS258           Y 1820-1439   74LS258 quad 2-to-1 line MULTIPLEXER
INV       74LS04            Y 1820-1199   74LS04 hex INVERTER
INV240    74LS240           Y 1820-1917   74LS240 Octal inverting bus buf/driver
INVSCH    74LS14            Y 1820-1416   74LS14 hex Schmitt-triggered INVERTER
JKFFPC    74LS109           Y 1820-1282   74LS109 dual JK-FF with Preset and Clear
NAND2     74LS00            Y 1820-1197   74LS00 quad 2-input NAND gate
NAND3     74LS10            Y 1820-1202   74LS10 tpl 3-input NAND gate
NAND4     74LS20            Y 1820-1204   74LS20 dual 2-input NAND gate
NOR2      74LS02            Y 1820-1144   74LS02 quad 2-input NOR gate
NOR3      74LS27            Y 1820-1206   74LS27 tpl 3-input NOR gate
N                           Y             7
            •               Y 1820-1          •
            •                                 •
            •                                 •
```

Modifying a Parts File

To modify an existing parts file, select Edit Parts File from the Manager menu. This causes the system to display a list of all parts files present on the system's mass storage devices. To modify an existing parts file, select the parts file from the menu displayed.

The system next displays the following options: ADD, CHANGE, DELETE, LIST, and SELECT ANOTHER FILE. Selecting one of these options allows you to modify the selected parts file. Use the left/right arrow keys, the delete character key, the RECALL key, and the insert character key when entering or modifying data for the parts file. The use of these keys is described in your computer's Operating Manual.

The function and use of each of the above options is described below.

- ADD - Selecting this option allows you to make additional entries to the parts file selected. Once the ADD option is selected the following lines are displayed:

```
Name     Type             LIST? STOCK #     Description
######## ################ # ########### ####################
```

  Enter the device name, electrical part type, and list option information in the appropriate fields (the stock number and description are optional) then press (CONTINUE). The width of each field is marked by the pound signs (#) appearing under each column heading. Continue entering the additional parts until you are finished. To exit the ADD branch, enter no information (press the CLEAR LINE key) and then press (CONTINUE). You may then either exit the Edit Parts File branch or select another option.

- DELETE - Selecting DELETE allows you to delete an existing entry from the parts file. When the DELETE option is selected the following lines are displayed:

```
Name     Type
######## ################
```

  Enter the device name and the electrical part type of the entry to be deleted and press (CONTINUE). The system next displays the information that you entered and gives you a chance to correct it before deleting the entry. If the system finds an entry in the parts file that matches the device name and electrical part type that you entered, that entry is deleted. Continue entering the device name and electrical part type of the entries to be deleted until you are finished. To exit the DELETE branch, enter no information (press the CLEAR LINE key) and then press (CONTINUE). You may then either exit the Edit Parts File branch or select another option.

- CHANGE - Selecting CHANGE allows you to modify any entry in the parts file. When the CHANGE option is selected the following lines are displayed:

```
Name     Type
######## ################
```

  Enter the device name and the electrical part type of the entry to be changed and press (CONTINUE). If the system finds an entry in the parts file that matches the device name and electrical part type that you entered, the entire entry is displayed. Make any changes that you desire to the entry and then press (CONTINUE). If the system does not find an entry in the parts file that matches the device name and electrical part type that you entered, an error message is displayed.

To continue modifying the parts file, again enter the device name and electrical part type of the entry to be changed. The last 15 entries changed since the CHANGE branch was selected are displayed at the top of the system CRT. To exit the CHANGE branch, enter no information (press the CLEAR LINE key) and press (CONTINUE). You may then exit the Edit Parts File branch or select another option.

Note

The CHANGE branch cannot be used to change either the device name or the electrical part type of a parts file entry. To change either or both fields it is necessary to first delete the entry, and then add a new entry for the device name and electrical part type.

- LIST - Selecting LIST allows you to obtain a listing of the parts file. You may obtain this listing on the system CRT or on the current printer. The listing may be sorted either alphabetically by device name and electrical part type or by stock number. Selecting the LIST option causes the system to display the following prompt:

```
List to printer (y/n)?
```

Responding Y, for yes, to this prompt causes the listing to be printed on the current system printer. Responding N, for no, causes the listing to be displayed on the system CRT. Next the system displays the following prompt:

```
Should I sort by part name (y/n)?
```

Responding Y, for yes, causes the listing to be sorted alphabetically by device name and electrical part type. Additionally, when sorting alphabetically you may specify a portion of the file to be listed. After responding Y for yes to the above prompt, the system prompts you for the device name with which to begin the listing. No entry that occurs alphabetically prior to the device name and type that you enter will be included in the listing. Making no response to this prompt causes the entire file to be sorted and listed.

Responding N, for no, causes the listing to be sorted by stock number and the entire parts file is listed. Since sorting by stock number is a fairly slow process you will probably use this option only when you need a list of the parts to construct your circuit.

To abort the listing at any time, press the special function key (k8).

- SELECT another file - Selecting SELECT causes the system to display a complete list of all parts files currently on line. You may then select another parts file to edit.

Backing Up and Copying a Parts File

To back up a parts file you should use the System Backup Routines in the File Utilities supplied with the system. To help the system recognize a parts file, a special character is added to the beginning of the parts file name by the system when the file is created. This will be important when: you are using the Copy File branch, the Purge Basic File branch, or the Rename Basic File branch of the File Utilities. Additionally, this information is important when you are attempting to address the file with BASIC language statements, such as COPY.

When entering the name of the parts file in response to a file utilities prompt, you will need to first press the following keys simultaneously:

CONTROL B

For example, suppose you were copying a parts file named TEST, using the Copy Basic File branch of the File Utilities. In response to the prompt "`Enter the name of the file...`" you would enter:

CONTROL B `TEST` CONTINUE

When addressing a parts file with a BASIC language statement you will need to add the following to the beginning of the file name:

```
CHR$(2)&
```

For example, if you were copying a parts file named TEST from one mass storage device to another, you would enter the following:

```
COPY CHR$(2)&"TEST:msus" TO CHR$(2)&"TEST:msus"
```
EXECUTE

Creating a Parts File

To create a new parts file, select Edit Parts File from the Manager menu and then select CREATE Parts File. The system next prompts you to enter the new parts file name. You may assign a name of up to 5 characters, consisting of letters (A-Z), numbers (0-9), and the underscore character (_). If you want the parts file to reside on a particular mass storage device, you may include a mass storage unit specifier (msus) with the file name. If no msus is supplied with the file name, the system places the parts file on the mass storage device containing the EGS-45 system software. For example, to specify that the parts file named TEST be created on an HP9885M flexible disc with a select code of 8 you would enter:

```
TEST:F8
```

The system next prompts you to enter the size of the parts file to be created, in records. The relationship between the parts file size (in records) and the number of entries in the parts file (where an entry is a device name, electrical part type, list option, stock number, and description) is defined by:

Number of records = Number of entries + 1

Each record in the parts file requires 90 bytes of disc storage. If there is not enough room on your mass storage media to create the new parts file, the system displays a warning and again prompts you for the size of the parts file. Since the size of a parts file cannot be expanded, it is a good idea to specify a parts file size that is larger than you expect to use. A good rule of thumb is to specify a size 20% larger than you anticipate needing.

The Material List

Selecting Material List allows you to generate a list of the electrical parts on a schematic drawing. This list may be sorted either by reference designator or by stock number. The listing may be sent to either the system CRT or the current system printer.

The material list is generated by comparing the device name and the electrical part type of each library part found in the drawing with entries of the specified parts file(s). If a match is found, then that part is listed in the material list. If the part is not found in a parts file, the system looks one nesting level deeper in the library part to see if it contains any devices whose device name and electrical part type match an entry in a parts file. This continues until the system finds a match and adds the part to the material list or until the system determines that the library part is not an electrical part.

It is possible that partial matches can be made. For example, if a drawing contains an instance of the library part RES but has no electrical part type assigned to it (and the parts file contains one or more entries for the device, RES) an error is generated. This error, along with several others, is discussed in the section detailing the material list report.

Generating a Material List

A material list may be generated for any drawing that has been created in the following way:

1. Lay out your schematic drawing with the Graphics Editor, making sure to comply with the conventions listed in the Drawing Schematics section of this chapter. This includes making sure that: each electrical part is represented by a named instance of a library part, each electrical part has a reference designator (iname) assigned to it, and each electrical part has an electrical part type (tname) assigned to it.
2. Use the ARCHIVE command in the Graphics Editor to create an archive file from the device file containing your schematic drawing.
3. Make sure that for each unique electrical part that occurs in your drawing, a corresponding entry exists in a parts file. A unique electrical part is defined as one having a unique combination of a library part name and electrical part type. For example, a resistor represented by the library part RES having the electrical part type 10K, defines a unique electrical part (a 10K ohm resistor). While your drawing may use many instances of the library part RES with an electrical part type of 10K, only one entry in a parts file is required.

To begin generating the material list, select MATERIAL LIST from the Schematic Drawing module menu. The system then displays a set of choices that consist of a listing of all parts files found on the systems mass storage devices, and three branches labelled: USE ALL-FILES, RE-START - SELECTING FILES, and DONE - SELECTING FILES. You must indicate to the system which parts file(s) 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, it appears underlined on your CRT. If you make a mistake and select a wrong parts file, simply re-select that file (the underline will disappear). To select all parts files listed, select USE ALL - FILES. To start the selection process over again, select RE-START - SELECTING FILES.

When you are finished selecting the parts files, select DONE -SELECTING FILES. The system next prompts you for name of the archive file created from your schematic drawing:

```
Enter the name of the archive file (null to pick new parts files)
```

Enter the file specifier identifying the archive file that was produced from your schematic drawing. If the archive file does not reside on the system volume EWSYS, you need to include a volume specifier. For example, assume that an archive file named TEST.ARC resides on the volume, EW1, while the current default volume is EWSCH. To specify the archive file you would enter:

```
EW1:TEST.ARC
```

or


```
EW1:TEST
```

(the system knows that it is looking for an archive file and appends the suffix, .ARC, if it is omitted).

The system reads the archive file and begins processing the data. Next the system prompts you for the name of the file in which you want to store the report:

```
Enter the name of the file for the output (PRINTER: for a listing)
```

The last response to this prompt is displayed on the screen as a default. Responding `PRINTER:`, causes the report to be printed on the current system printer. Clearing the line and responding with no file name causes the system to back up to the last step and again prompt you for the name of the archive file created from your drawing. Any other response to the prompt causes the system to create a BASIC data file on the mass storage device containing the current system volume. To specify a mass storage device other than the one containing the current system volume, you may add a msus to the file name that you enter. For example, entering:

```
EXAMP:F8
```

causes the system to place the material list in the BASIC data file EXAMP, located on an HP 9885 disc drive with an interface select code of 8.

The system next asks you whether you want the material list sorted by reference designator or by stock number:

```
Should the list be sorted by Reference designator
or Stock number? (R/S/CONT)
```

Responding `R` causes the list to be sorted by reference designator. Responding S causes the list to be sorted by stock number. No response (pressing only **CONTINUE**) causes the system back up and again prompt you for the name of the output file. The format of both types of report is discussed in the next section of this chapter.

The system next prompts you for the number of lines per page that you want in the report. The number you supply is exclusive of the number of lines required by titles and headers.

```
How many data lines do you want per page?
```

The last response to this prompt automatically appears on your screen as the default value. Entering no response causes the system to back up and again prompt you as to whether you want the list sorted by stock number or by reference designator.

The next prompt by the system allows you to enter a one line (50 character maximum) page header for your report.

```
Enter the one line header (50 characters maximum)
```

Making no response to this prompt causes no header at the top of the report.

If you have specified that the report generated is to be printed on the current system printer, the system makes the additional prompt:

```
How many copies do you want?
```

The last response to this prompt automatically appears on your screen as the default number of copies. Entering no response causes the system to back up and again prompt you for the number of lines per page.

Format of the Material List

The material list report, regardless of whether it is sorted by stock number or by reference designator, has a header with the following format:

```
                         Material List
                      <archive file name>

iname (if the system's library part is used)
tname (if the system's library part is used)

<one line header>
                                           <today's date> PAGE n
```

The date referenced above is the date entered when first loading the EGS software. Each page of the material list report generated has a page number appearing in the upper right hand corner of the page.

If the material list is sorted by reference designator, the body of the report has the format

```
REFERENCE      STOCK NO.      DESCRIPTION
-------------  -------------  -----------------------------------
```

where:

- REFERENCE is the reference designator assigned as the iname of the instance of the library part. All instances with the same class designator (first letter of the iname) and the same unit number (second letter of the iname) will be listed together. For example, the reference designators U1A, U1B, U1C, and U1E will be listed as: U1A-C,E.
- STOCK NO. is the stock number associated with the parts file entry.
- DESCRIPTION is the description associated with the parts file entry whose device name and electrical part type matches the library part name and the tname of the library part representing the device being listed.

If the material list is sorted by stock number, the body of the report has the format

```
STOCK NO.      QTY   DESCRIPTION          REFERENCE
------------- ----- -------------------- ----------------------
```

where:

- STOCK NO. is the stock number associated with the parts file entry whose device name and electrical part type matches the library part name and the tname of the instance representing the electrical device being listed.
- QTY is the quantity of parts with the corresponding stock number occurring in the schematic drawing.
- DESCRIPTION is the description associated with the parts file entry whose device name and electrical part type matches the library part name and the tname of the library part representing the device being listed.
- REFERENCE is the reference designator assigned as the iname of the instance of the library part. All instances of library parts with the same class designator (first character of the iname) and the same unit number (second character of the iname) will be listed together. For example, the reference designators U1A, U1B, and U1C will be listed as: U1A-C.

Material List Error Checking

The program that generates the material list is capable of catching and flagging several types of errors. For example, a library part missing a reference designator or a library part missing an electrical part type is flagged as an error. These errors cause a special character(s) to be added to the material list report in one or more of the fields of the report. A list of these character(s) and their possible meanings is provided below.

The asterisk (*) is used in the material list report to indicate that a value is missing. It is generated under the following situations:

- When an instance of a library part, whose device name and electrical part type (tname) match the device name and type of a parts file entry, is missing a reference designator (iname). An asterisk (*) is placed in the reference designator field in the material list report.
- When an instance of a library part has no electrical part type (tname) assigned to it and sorting by reference designator is specified. An asterisk is placed in the reference designator field, while the name of the library part missing the part type is listed with an error message in the description field.

The blank field is used in the material list report to indicate that:

- A value is missing or that no corresponding entry in the parts file exists for that library part.
- An optional field (such as the description or the stock number) in the parts file entry has been left blank.

  Since every library part that exists in your schematic drawing is considered to be an electrical part, an entry in the parts file is required for every library part used in the drawing. Use the list option to specify that a part is not to be included in the report. For example, if you use a library part to represent a ground symbol in your schematic drawing, an error is generated if an entry for this part does not exist in a parts file. Failure to do this causes a blank to be listed in one or more fields of the output.

The parentheses ( ) are used in the material list report to:

- flag duplicate reference designators that occur in the material list.

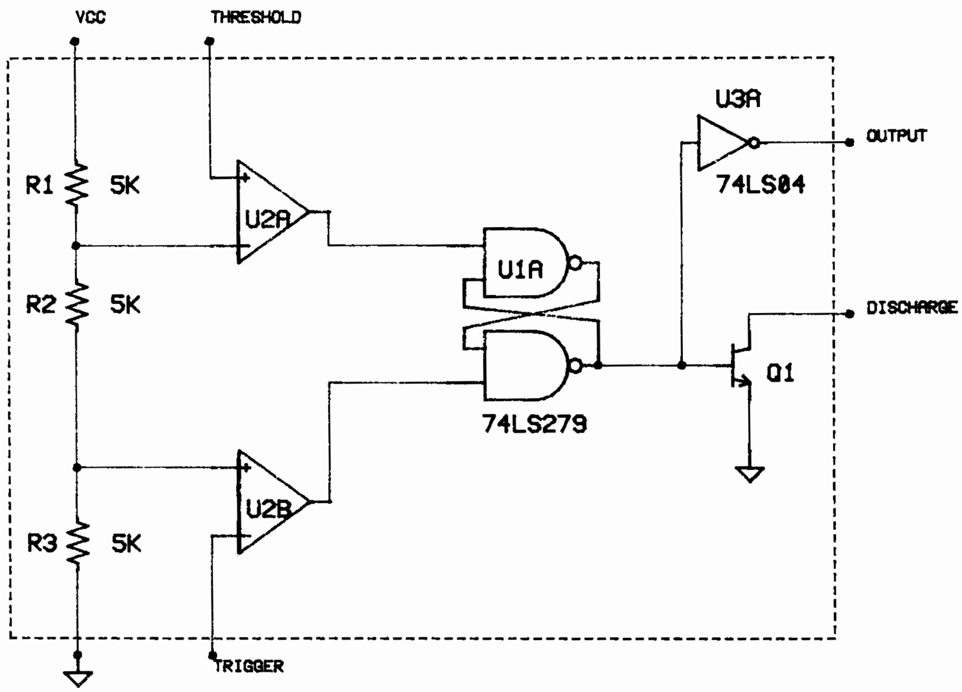
  When a material list is sorted by reference designator, any repetition of reference designator in the list is enclosed in parentheses. When a material list is sorted by stock number, only duplicate reference designators assigned to the same stock number are enclosed in parentheses.

Modifying the Material List Format

The material list program is a BASIC language program stored on your mass storage device in the file, MATLST. The program is fully documented to help you understand the existing code. A thorough working knowledge of BASIC is suggested before attempting to modify the format of the report.

Schematic Layout Example

This section lists the steps to build the schematic shown below using the EGS-45 Schematic Drawing Module. Remember that this is only one of many possible methods of drawing schematics. You can change it to suit your own needs.



1. Load EGS-45 and enter the Schematics Module.
2. Pick "EGS-45 Graphics Editor" from the Schematics Layout section of the main menu.
3. You are now in the Graphics Editor, which has been customized (via menu, process, and macro files) for schematics layout. Now let's draw the schematic.

   a. Enter the Edit subsystem
   ```
   EDIT;
   ```
   b. Snap to every .25 cm, and when the grid is displayed, display every second grid point
   ```
   GRID .25,2;
   ```
   c. Turn on display of the grid
   ```
   GRID;
   ```
   d. Set scaling to 1:1 and set the nesting level to 3
   ```
   WINDOW $XMAG 1 $NEST 3;
   ```
   e. Add the parts
   ```
   ADD SRLATCH2 0.00, 0.00;
   ADD INV 3.5, 1.5;
   ADD NPN 4, -1.75;
   ADD OPAMP -3.5, 1; -3.5, -3.25;
   ADD RES $ROTCCW -5.5, .5; -5.5, -1.25; -5.5, -4.75;
   ```

f. Connect the parts
```
CONNECT 4.75, 1.5 6, 1.5; 4.5, -1.25 4.5, -1
6, -1; 4.5,
-2.25 4.5, -3; 2.25, -1.75 4, -1.75; -1.75, -3.75 -1.75, -2
0, -2; -1.75, .5 -1.75, 0 0, 0; -5.5, 3 -5.5, 1.25; -5.5,
.5 -5.5, -.5; -5.5, -1.25 -5.5, -4; -5.5, -3.25 -3.5,
-3.25; -5.5, -4.75 -5.5, -6; -3.5, -4.25 -3.5, -6; 3.5,
-1.75 3.5, 1.5; -3.5, 3 -3.5, 1; -5.5, 0 -3.5, 0;
```

g. Add the junctions
```
JUNCTION 3.5, -1.75; 6, 1.5; 6, -1; -5.5, 0; -5.5, 3;-3.5,
3;-5.5, -3.25; -3.5, -6; -5.5, -6; 2.25, -1.75;
```

h. Add the grounds
```
ADD GND -5.5, -6; 4.5, -3;
```

i. Add the outline
```
OUTLINE -6.5, -5.75 -6.5, 2.75 5.75, 2.75 5.75, -5.75 -6.5,
-5.75;
```

j. Add some notes
```
ADDNOTE 'VCC' -5.5, 3.25; 'THRESHOLD' -3.5,3.25; 'OUTPUT'
6.25, 1.5; 'DISCHARGE' 6.25, -1; 'TRIGGER' -3.5, -6.25;
```

k. Add the reference designators
```
ADD_REF U1A 0, -2.25 .75, -.5;
ADD_REF U3A 4, 1.5 4, 2;
ADD_REF Q1 4, -2.25 4.75, -2;
ADD_REF U2A -3.5, -.25 -3, .25;
ADD_REF U2B -2.75,-3.75 -3, -4;
ADD_REF R1 -5.5, 1 -6.25, .75;
ADD_REF R2 -5.5, -.75 -6.25, -1;
ADD_REF R3 -5.5, -4.25 -6.25, -4.5;
```

l. Add the types
```
ADD_TYP 74LS279 0,-2.25 .5, -2.75;
ADD_TYP 74LS04 4, 1.5 4, .75;
ADD_TYP 5K -5.5, 1 -5, .75;
ADD_TYP 5K -5.5, -.75 -5, -1;
ADD_TYP 5K -5.5, -4.25 -5, -4.5;
```

m. Save the drawing on volume EWSCH, and call it EXAMP
```
SAVE EWSCH:EXAMP;
```

n. Plot the schematic 1:1
```
WINDOW $FIT; PLOT :S1;
```

o. Leave the Edit subsystem
```
EXIT;
```

p. Create an Archive file of all layers and maximum nesting from the drawing
```
ARCHIVE EWSCH:EXAMP EWSCH:EXAMP;
```

q. Leave the Graphics Editor
```
BYE;
```

4. Now pick "Material List" from the main menu.
5. Pick "USE ALL - FILES" from the Material list menu. This causes the material lister to search all parts files that are on-line. This example was created with the two supplied parts files (SPART & GPART) in mind. The material lister will prompt you for the rest of the information it needs. Try all the possibilities to become familiar with the operation of the material lister.

Material List
EXAMP

Hewlett-Packard Company 24-DEC-81 PAGE 1

REFERENCE	STOCK NO	DESCRIPTION
Q1		NPN transistor
R1		(RES 5K Is not in the parts file).
R2		(RES 5K Is not in the parts file).
R3		(RES 5K Is not in the parts file).
U1A	1820-1440	74LS279 quad SR LATCH
U2A,B		Operational amplifier
U3A	1820-1199	74LS04 hex INVERTER

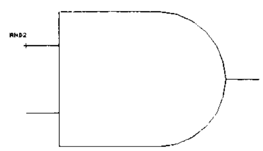
Material List
EXAMP

Hewlett-Packard Company 24-DEC-81 PAGE 1

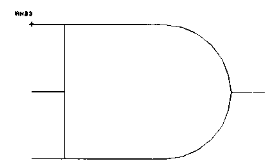
STOCK NO	QTY	DESCRIPTION	REFERENCE
	1	(RES 5K Is not in the parts file).	R3
	1	(RES 5K Is not in the parts file).	R2
	1	(RES 5K Is not in the parts file).	R1
	1	Operational amplifier	U2A,B
	1	NPN transistor	Q1
1820-1199	1	74LS04 hex INVERTER	U3A
1820-1440	1	74LS279 quad SR LATCH	U1A

The following section contains representations of the supplied library parts.

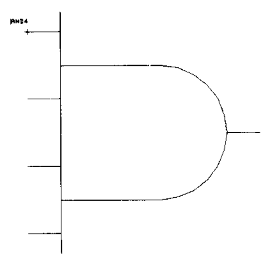
AND2



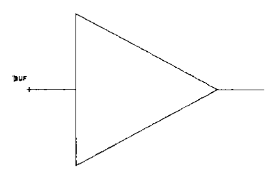
AND3



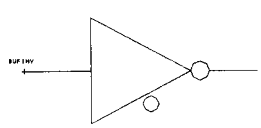
AND4



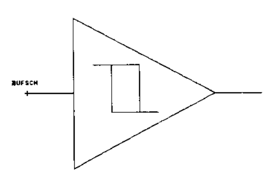
BUF



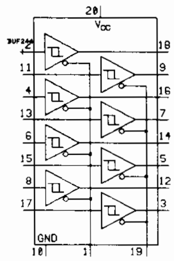
BUFINV



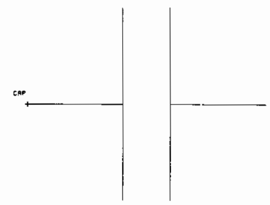
BUFSCH



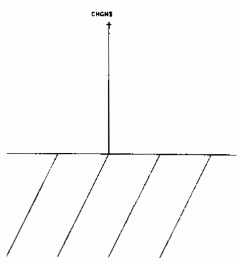
BUF244



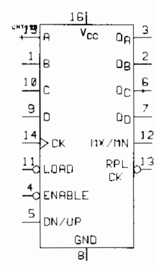
CAP



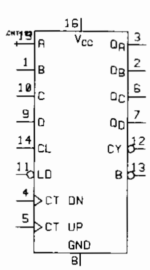
CHGND



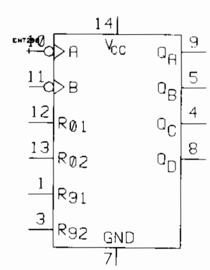
CNT190



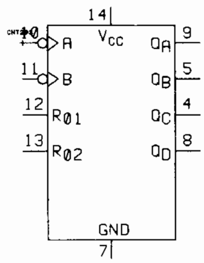
CNT193



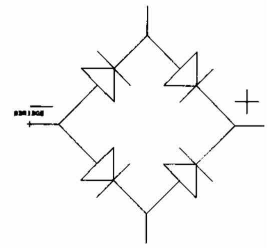
CNT290



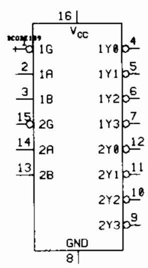
CNT293



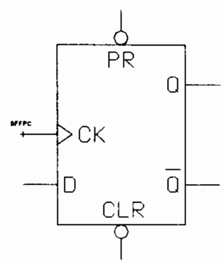
DBRIDGE



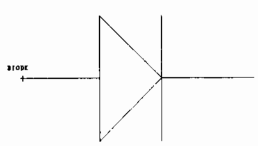
DCODE139



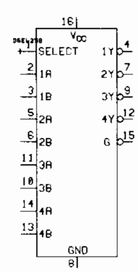
DFFPC



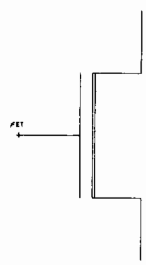
DIODE



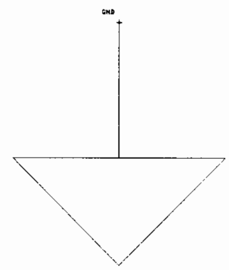
DSEL258



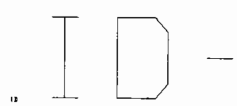
FET



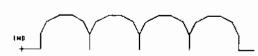
GND



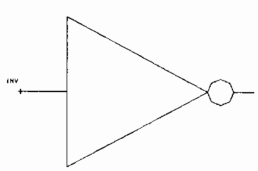
ID



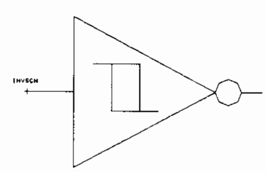
IND



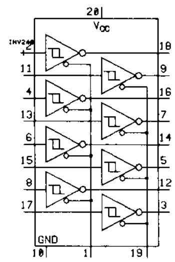
INV



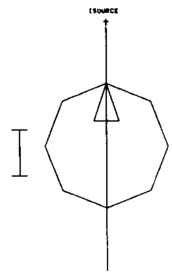
INVSCH



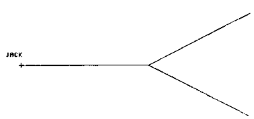
INV240



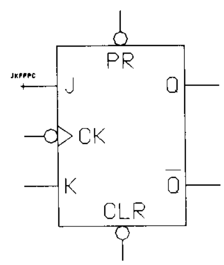
ISOURCE



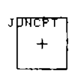
JACK



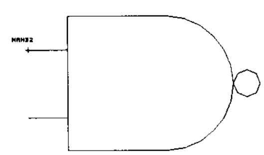
JKFFPC



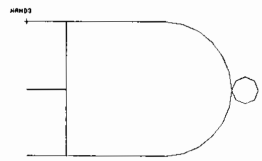
JUNCPT



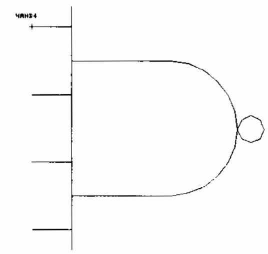
NAND2



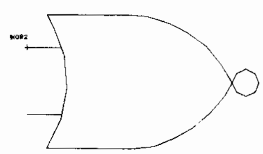
NAND3



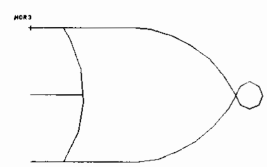
NAND4



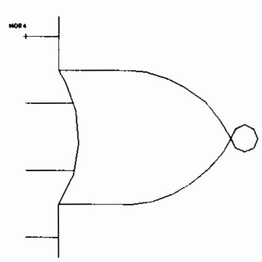
NOR2



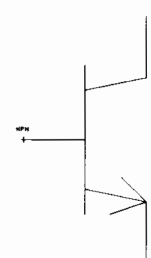
NOR3



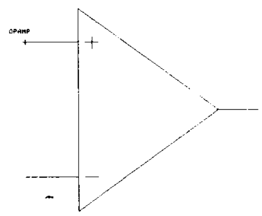
NOR4



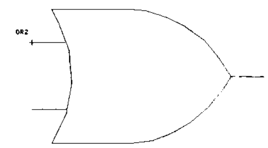
NPN



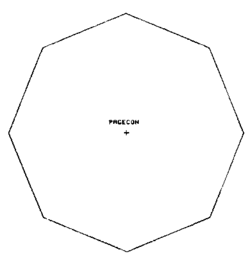
OPAMP



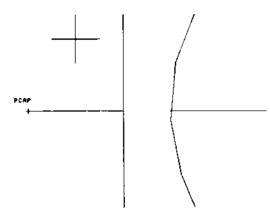
OR2



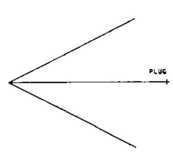
PAGECON



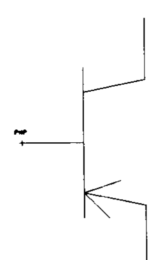
PCAP



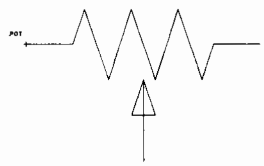
PLUG



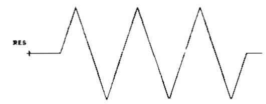
PNP



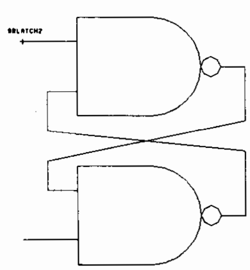
POT



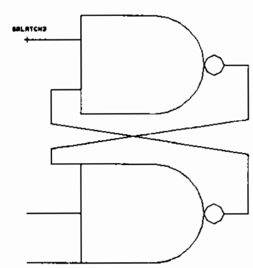
RES



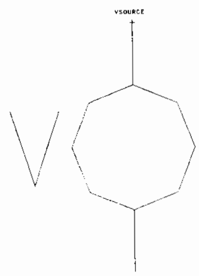
SRLATCH2



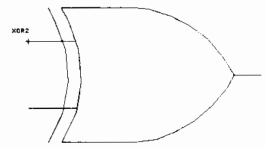
SRLATCH3



VSOURCE



XOR2



ZDIODE

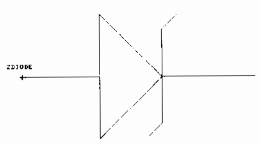


Table of Contents

Chapter 8
P. C. Board Layout Module

Overview

The P. C. Board Layout Module is an application module for EGS-45. It provides the process, macro, and menu files which customize the EGS-45 Graphics Editor for the layout of printed circuit boards. Additionally, a set of library parts are provided to help you lay out your P. C. board. You may use these parts in your drawing or use them as examples to pattern the design of your own library parts.

An application program supplied with this module produces Excellon format data for an N/C drill machine. The N/C data is prepared directly from the P. C. board drawing that you create with the Graphics Editor. Additionally, an application program is provided which produces the instructions to drive a Gerber 4300/4400 series photo plotter. With knowledge of the BASIC programming language, these application programs can be modified to produce different formats for other photo plotters and N/C drill machines.

The following items are shipped with the P. C. Board Layout module:

- P. C. Layout Module Manual (HP part# 98303-12111)
- P. C. Layout Module software cartridge (HP part# 98303-12504)

A printed circuit board is created on this system in the following way:

1. Create a drawing that represents your printed circuit board using the Graphics Editor and its subsystems. The conventions of the supplied application programs must be followed if N/C drill tapes are to be produced and if board artwork is to be produced on a Gerber photo plotter. Save the drawing on an EW volume.
2. Create a generate file from the device file containing your P. C. board drawing. Use the PLOT command in the edit subsystem to create a verification pen plot.
3. Create a virtual punch file from the generate file created in step 2 above.
4. Create data for an Excellon format N/C drill machine by selecting the appropriate function from the Manager menu. The data is created from the virtual punch file created in step 3 above.
5. Create a virtual plot file from the generate file created in step 2 above.
6. Create data for your 4300/4400 series Gerber photo plotter by selecting the appropriate function from the Manager menu. The data is created from the virtual plot file created in step 5 above.

A printed circuit board is created with this module by using the Graphics Editor to lay out components on the CRT. These components represent the pattern of copper to be deposited on the finished board (this may vary with fabrication shop procedures and photo plotter film type used). These patterns are plotted on clear film (again this may vary with the fabrication shop procedures) with all plotted items (wherever the film is not transparent) becoming copper. The photo plotter programs allow you to fill certain components instead of merely outlining them. This allows you to add solid areas of copper to the board.

The maximum complexity of a printed circuit board that can be designed and laid out with this module depends on the amount of memory in your computer and the techniques you use to lay out the board. A typical board created with this module is approximately 80 square inches with a density of 40 integrated circuits. Later sections in this chapter deal with creating larger and/or more complex P. C. boards.

This application module provides a set of tools to help you design and layout P. C. boards in an efficient manner. This manual teaches you methods of applying these tools to the methods of P. C. design that you already know.

Before continuing with this chapter, you should make sure that you read and understand the manual for EGS-45. Knowledge of the Graphics Editor and its subsystems is required before a P. C. board may be created with this module. Additionally, you should understand and be able to use the capabilities of the File Utilities module. A good working knowledge of the BASIC programming language may also be required to direct output to a device other than the devices directly supported by this module (such as a tape punch other than the HP 9884 tape punch or an N/C drill machine). Knowledge of the BASIC programming language is also required to modify the format of the output to a device. For example, you need to know the BASIC programming language to modify the photoplotter output to drive a photo plotter other than a Gerber 4300/4400 series photo plotter.

Structure

The concept of customizing the software to meet the needs of a specific application is discussed in chapter 6. In that chapter you learned to create your own application modules by writing BASIC application programs and creating control files that could be recognized by the system. The P. C. Board Layout module is nothing more than a combination of the techniques discussed in that chapter and chapter 2. This application module provides a process file that is customized for laying out P. C. boards. Additionally, a set of library parts is provided which further customizes the module for the application. Finally, special BASIC language application programs are provided that allow you to direct your finished artwork to a photo plotter. These application programs may be selected from the Manager menu.

The system's software diagram located at the beginning of chapter 1 shows how the P. C. Board Layout module interacts with the rest of the EGS-45 software.

Preparing the Software for Use

The P. C. Layout module software is supplied on magnetic tape cartridges. Since this type of mass storage is not supported by the Engineering Graphics System, it is necessary to transfer the programs and files from the tape cartridge to a supported mass storage medium. The BASIC language program COPY is provided on the software cartridge to aid you in this transfer. To run the copy program, do the following.

1. Reset the computer to its initial state by either simultaneously pressing (CONTROL) (STOP) or by turning the computer off and then turning it on again.
2. Insert the P. C. Layout module software cartridge (HP part# 98303-12504) into the right hand tape drive of your computer).
3. Make sure that the disc in your mass storage device has been initialized. The disc may contain other files, but should have at least 200,000 bytes of storage space available.
4. Enter the following command on your computer:

   `LOAD "COPY:T15",1` (EXECUTE)

5. The computer next prompts you for the msus (mass storage unit specifier) of the source device. This is the mass storage device containing the software to be copied. You should respond by entering:

   `:T15` (CONTINUE)

6. Next you are prompted for the msus of the destination device. This is the mass storage device to which you wish transfer the software. Enter the appropriate msus and press (CONTINUE).
7. Next the system displays the source and destination msus that you entered. It then prompts you to indicate if these msus are correct. Responding that the msus are not correct causes the system to again prompt you for the source and destination msus.
8. Press (C) to initiate the software transfer of all files on the source device to the destination device.*

   Your disc now contains several BASIC files and the EWE volume EWPC. EWPC contains the macro, menu, process, and device files (library parts) that you will use to create printed circuit board artwork. Before attempting to create photo plotter data or N/C data from your P. C. drawings, use the System Configuration module to add the devices that you will use (such as the HP 9884 or HP 7970E) to the system configuration. The tape punch **must** be added to slot 9 in the peripheral table. The HP 7970E **must** be added to slot 10 in the peripheral table.

* Should the copy program encounter an existing file with the same name as the file being copied (as might occur when copying a new software release to the disc), you have three options.
1. Purge the existing file before making the copy or do not copy.
2. Automatically purge all existing files before performing the copy.
3. Skip duplicate files and continue with the next file.
The options are selected by pressing C, P or S, as instructed by the program.

Concepts

The Graphics Editor

Selecting the Graphics Editor branch from this module allows you to lay out your board design on the graphic CRT. Boards are laid out using the commands of the edit subsystem. A customized process file has been provided that defines the layers and units for the specialized artwork incurred with P. C. board layout. This process file is stored in the EW volume EWPC with the process file name of PC.PRO. When the Graphics Editor is selected from this module, the process file named PC is automatically loaded into the system. To automatically load your own customized process file, the process file name should be PC. Additionally, you may modify the module's control file to change the name of the process file which is automatically loaded.

The macro and menu files PC.MAC and PC.MEN are also automatically loaded into the system when the Graphics Editor is selected from this module. Again, to automatically load your own customized menu and/or macro files, their file names should be PC. By modifying the control file for this module, you may change the name of the macro and menu files which are automatically loaded.

Logical Layers vs. Physical P. C. Board Layers

Logical layers are the 255 layers that compose the drawing area on the CRT. The drawing area may be thought of as 255 overlaid sheets of clear plastic. These layers may be viewed individually or in combination with other layers. Only layers defined in the process file may be drawn on and only layers turned on with the SHOW command may be displayed and/or modified.

Physical board layers (also called board sides) are the actual physical layers of copper and fiberglass that make up a single layer of a multi-layered board. Each physical layer referred to in this manual represents a layer of copper or of solder resist. Information on one logical layer may be used to construct artwork for several physical board layers. Similarly, the artwork for one board layer may be so complex that several logical layers are needed to draw the artwork. Logical layers can be combined when producing plots.

By creating a new process file, using the existing process file, or modifying the existing process file you control the relationship between logical layers and physical layers. The EGS-45 software requires that logical layer zero (0) be reserved for system use and requires that it be defined in all process files. Also, HP reserves layers 1-100 for use by factory defined process files.

Working With a Photo Plotter

The P. C. Board Layout module provides for automatic creation of Gerber photo plotter data. With some modification, this capability can be extended to other photo plotters. Photo plotters differ from pen plotters (such as the HP 9872C or the HP 7850) in that they shine light on light sensitive films to produce a plot on photographic film. Pen plotters use ink pens to produce the plot on paper or sheet plastic.

Gerber photo plotters produce their plots by shining light through one of 24 specially shaped holes, called apertures. Apertures come in a variety of shapes (such as ovals, squares, circles, etc.). These apertures are located on a wheel in the photo plotter, called the aperture wheel. If the component to be plotted has the same dimensions as one of the apertures, the photo plotter can simply position the aperture wheel in the correct position and plot and fill the component by flashing light through the aperture once. This process is called **flashing**. If the component to be plotted does not have the same dimensions as one of the apertures, the photo plotter selects an aperture, and while shining light through it, moves the aperture such that the beam of light outlines and fills the component on the light sensitive film. This process is called **tracing**.

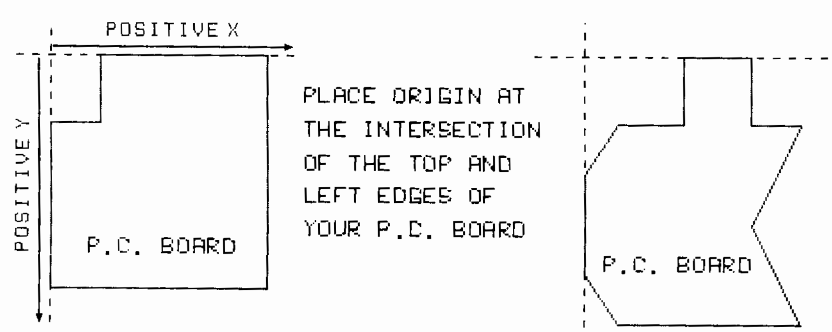
Parameters specified when creating a virtual plot file and creating photoplotter data determine which components are to be traced and which are to be flashed. You may also specify that a component is not to be plotted on the photo plotter at all or that a component is to be flashed with a specified aperture. These options are discussed in the sections dealing with these functions and the conventions that they use.

Conventions of P. C. Board Layout

If you intend to use the photo plotting and N/C drill programs supplied with this module, certain conventions of P. C. board layout must be followed. These conventions are mentioned here so that you may allow for them when laying out your board. The specifics of each convention are discussed in the sections of this chapter that deal with photo plotting and N/C drill code generation.

The N/C drill code generation programs assume that drilled holes are represented as circles. The location of the circle's center identifies the location of the hole to be drilled. The size of the hole is determined by the iname assigned to the circle. The iname must have the format Dxxxx, where xxxx is the finished hole size, in mils (1 mil =.001 inch).

The N/C drill code generation programs require that a special circle with the iname .ORIGIN be added to every layer which contains drill code information. This circle must be at the intersection of the upper and left edges of the board outline; in other words, the upper left hand corner of the smallest rectangle that could totally enclose the board (see the drawing below). The circle does not define a physical hole, rather it simply identifies a reference point for the N/C drill code generation programs. The requirements for the placement of these special components are discussed in greater detail in the sections of this chapter that deal with the N/C drill code generation programs.



Assigning an iname of the form .Zxxxxxx to a circle or rectangle component causes the component to flash instead of being traced when photo plotted. The aperture that is used to flash the component is identified by the characters xxxxxx in a table called the Gerber Aperture Table (or wheel table). For example, assigning the iname, .ZW10, causes the component to be flashed using the aperture identified in the Gerber Aperture Table by the name, W10. The Gerber Aperture Table is discussed in greater detail in the section dealing with the conventions required by the photo plotter programs.

The Process File

The process file defines the relationship between logical layers of the system and the physical P. C. board sides or layers. Logical layer zero, as always, is reserved for displaying instance components. All of the graphical information on the logical layers represents features of the board as viewed from the top or component side of the board. When drawings are sent to the photo plotter via the programs supplied, all lines are drawn with a solid line type, regardless of the line type defined in the process file. This allows you to use dotted lines, broken lines, and dashed lines to visually separate certain parts on the CRT display without having these line types show up on your photo plotted artwork. Outside of these limitations, you may define the process file for the P. C. Layout module in any way that you deem necessary. Remember that HP reserves layers 1-100 for factory use.

To help you understand how the process file can ease your P. C. board layout task, several examples are provided. In the following section, a simple process file is created for designing a four layer P. C. board. Techniques of board layout are then considered and the simple process file is enhanced to accomodate the techniques discussed.

A Simple Process File

A simple process file that could be used to lay out a four layer P. C. board might utilize one logical layer for each physical board layer plus two logical layers for drilled hole information. An example of this simple process file is described below.

Logical Layer	Label	Description
0	INSTANCE	This layer is reserved by the system to display instances of library parts.
1	PDRILL	This layer contains the circle component with iname ORIGIN, and all circles representing plated drilled holes.
23	UPDRILL	This layer contains the circle component with iname ORIGIN, and all circles representing unplated drilled holes.
5	COMPONET	This layer contains components that represent all traces, pads, and planes (or any other component representing metalization) on the component side of the board.
11	INLAYRA	This layer contains components that represent all traces, pads, and planes (or any other component representing metalization) on the inner physical board layer, A.
12	INLAYRB	This layer contains components that represent all traces, pads, and planes (or any other component representing metalization) on the inner physical board layer, B.
6	CIRCUIT	This layer contains components that represent all traces, pads, and planes (or any other component representing metalization) on the circuit side of board.

Photo plots of the individual physical board layers are obtained by combining logical layers (those that represent physical board layers) when creating a virtual plot file. By then creating photoplotter data from the virtual plot file(s), photo plots may be generated on a Gerber series 4300/4400 photo plotter. These photo plots may then be turned into P. C. boards through the magic of the fabrication shop.

Expanding the Simple Process File

If your P. C. board requires a solder resist mask for each outer layer of the board, you might add two more logical layers to your process file definition. The process file might be updated to include the following:

Logical Layer	Label	Description
21	SLDRCOMP	This layer contains components that represent areas in which you do not wish solder resist on the component side of the board.
22	SLDRCIRC	This layer contains components that represent areas in which you do not wish solder resist on the circuit side of the board.

The two logical layers, SLDRCIRC and SLDRCOMP, both contain components that represent areas in which NO solder resist is desired. For this reason, these logical layers are referred to as negative areas, or keepout areas. This contrasts with the logical layers that represent the physical sides of the board where components in the logical layers represent areas in which copper **is** desired. These logical layers are referred to as positive areas. Photo plots of these two logical layers may be obtained by creating a separate virtual plot file for each logical layer.

While using the process file above would allow you to obtain the artwork necessary to create a four layer board, it causes you to lay out each indvidual component on each layer just as it is on the physical board layer. This works well for pads, traces, and planes that are unique to an individual layer. However, for those layers that require pads, traces, or other metalization in identical locations (such as pads on the circuit side **and** the component side for plated through holes) extra time is required to make the identical entry on each layer.

If you were to add a logical layer definition to the process file for entry of components common to multiple logical layers, it would only be necessary to add the common component to the drawing **once** (in the common logical layer). Then the contents of the common logical layer could be added to each individual logical layer to obtain the total physical board layer, when creating the virtual plot files.

For example, suppose you create a logical layer that contains all pads that are to appear on both the component side of the board and the circuit side of the board. You enter those pads that appear on both sides of the board in the common, or master, logical layer once. Then enter those pads, traces, and planes that are unique to an individual board side on the individual logical layer representing that board side. The final plot for the component side of the board is obtained by creating a virtual plot file which contains both the master logical layer and the COMPONET logical layer. By creating a photoplotter data from the virtual plot file, the finished photo plot of the component side of the board is obtained. The photo plot of the circuit side of the board may be obtained in a similar fashion. Therefore, by simply adding another logical layer you have eliminated the time required to enter those common components on each individual logical layer.

To incorporate the use of common logical layers, additional layer definitions may be added to the process file above. Two such logical layers look like this:

Logical Layer	Label	Description
27	FOTOPADM	This logical layer contains components that represent areas that are to become copper on **both** the component and circuit sides of the board for photo plotting.
28	FOTOSLDM	This logical layer contains components that represent areas that should not be covered with solder resist on **both** the component and circuit sides of the board for photo plotting.

Similarly, we can create a logic layer that contains components representing areas that should not be covered with solder resist, which are optimized for display on the CRT.

Logical Layer	Label	Description
30	DISPSLDM	This is the solder resist keepout master layer when using the graphics CRT (i.e., when editing).

If you look back at the definitions of layers 27 and 28, you will see that they are optimized for photo plotting. You can use the SHOW command in the Graphics Editor to turn on the display specific while you are using the CRT. If you want to do artwork with fill on a pen plotter, you again can use the SHOW command to turn on the pen plot specific layers and turn off the display and photo plot specific layers.

Since most P. C. shops require that a pen plot of the board outline (called a board blank) and tooling holes be provided for the router, a separate logical layer might be defined:

Logical Layer	Label	Description
2	BDBLANK	This logical layer contains components that represent the P. C. board outline and the tooling hole pads for the router.

Since each physical board layer requires registration marks, a separate logical layer can be defined that contains nothing but registration marks and other information needed by the fabrication shop (such as text, labels, etc). The information contained in this logical layer may be added to the information contained on the other individual logical layers when creating the virtual plot file. This logical layer is defined as follows:

Logical Layer	Label	Description
19	REFRENCE	This logical layer contains components that represent the registration marks and text for the P. C. fabrication shop.

Many P. C. shops require assembly drawings. These drawings contain outlines of parts.

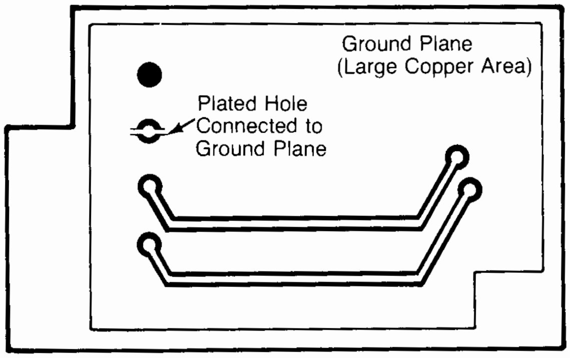
Logical Layer	Label	Description
26	ASMBDRAW	This logical layer contains outlines of parts for assembly drawings.

Extending the Simple Process File to Handle Planes

When creating multi-layer P. C. boards, large areas of copper are often desired to provide a ground or power plane. You could create these planes by adding components to the logical layer representing the particular board side. However, because certain component shapes may be filled when photo plotting, you may not be able to obtain the shape of the plane that you want. An alternate method of creating solid planes on a P. C. board calls for the definition of additional logic layers and the use of photographic reversal and combination techniques.

Figure A

Inner Physical Board Layer A



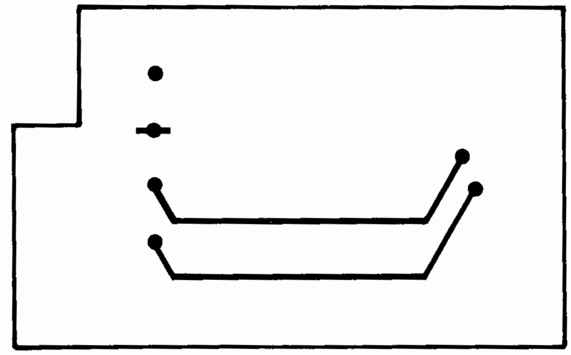
Areas Void of Copper (Keepout Areas)

Copper Trace

Copper Pad

Figure B

Photo Plot of Logical Layer, INLAYRA

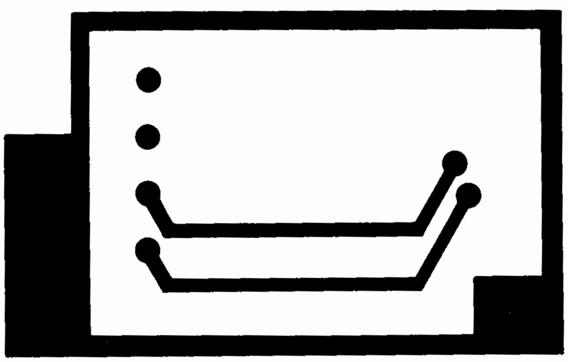


Areas to Become Copper

Areas Void of Copper (Keepout Areas)

Figure C

Photo Plot of Logical Layer, INKOA

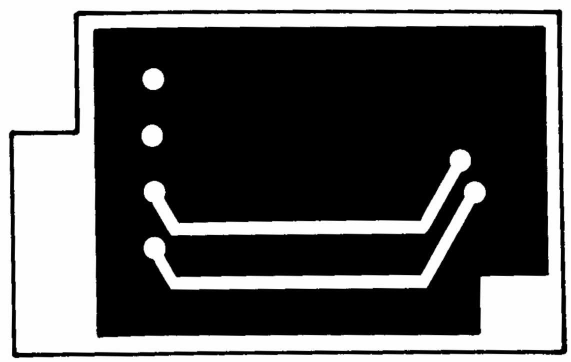


Areas Void of Copper (Keepout Areas)

Areas to Become Copper

Figure D

Photographic Negative Image of Figure C

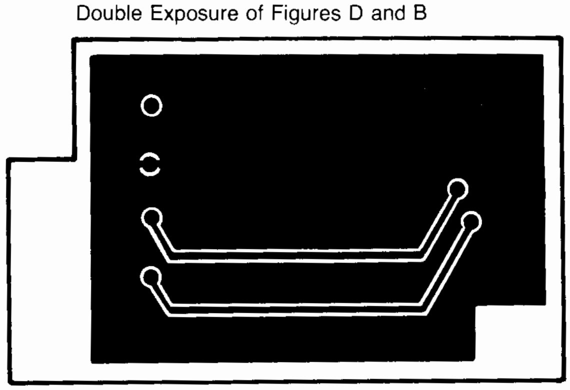


Areas to Become Copper

Areas Void of Copper (Keepout Areas)

Figure E

Double Exposure of Figures D and B



Areas to Become Copper

Areas Void of Copper (Keepout Areas)

Consider the inner board side, inner layer A, represented by figure A in the following set of drawings. It consists of copper pads and traces, keepout areas (areas in which NO copper is desired) around plated through holes and traces, and a large ground plane that is formed by the large copper area occupying the remainder of the board layer. Traces and pads for plated through holes may be created by adding the appropriate components to the logical layer, INLAYRA, that represents the inner physical board layer, layer A. This represents no variation from the methods discussed previously. The photo plotted output of this logical layer is shown in figure B in the set of drawings below.

However, to add the ground plane, a new logical layer must be defined:

Logical Layer	Label	Description
15	INKOA	Keepout areas for inner physical layer A.

Components added to this logical layer represents areas in which NO copper is desired. For this reason, this logical layer is called a keepout layer (also called a negative layer). What this says, in essence, is that you desire the entire inner layer A to be copper except for the areas in which components are placed. Keepout areas for plated through holes and keepout areas around traces should be placed in this logical layer, as shown in the drawing. Pads for plated through holes and traces should be represented by the appropriate components placed in the positive logical layer (where components represent metalization), INLAYRA. By photo plotting the negative logical layer, INKOA, you receive a plot in which plotted components represent area NOT to be metalized on the finished P. C. board (see figure C in the set of drawings below).

By photographically producing a negative image of the photo plot of the keepout area (see figure D in the set of drawings below) a film representing areas to become copper is produced. If this plot is then photographically combined with the photo plot of the traces and pads of the inner layer (by means of double exposure or some other method of combination) a final photo plot representing the inner physical layer is obtained (see figure E in the set of drawings below). In this plot, dark areas represent areas to become copper while light areas (clear areas on the photo plot) represent areas to be void of copper. This plot can then be used by the fabrication shop to produce the finished board.

As you have seen, by defining the additional logical layer to represent keepout areas on a physical board layer, you are able to add irregularly shaped plane areas to your P. C. board design. This insures that traces and pads are isolated from the plane area. This concept of keepout areas may be extended to the other inner physical layer by defining an additional logical layer in the process file. Adding the keepout layer for the inner physical board layer, B:

Logical Layer	Label	Description
16	INKOB	Keepout areas for inner physical layer B.

These individual keepout logical layers provide sufficient information to produce multi-layer P. C. boards with irregularly shaped plane areas. However, it requires that components for the keepout areas be added to each individual keepout logical layer. This works well for keepout areas that are unique to each layer. However, for keepout areas that are common to multiple layers (for example, plated through holes require identical keepout areas on each physical board layer) an easier method is available. This method involves defining another logical layer to act as a master keepout logical layer.

Components representing keepout areas common to multiple board layers are added to the keepout master while keepout areas unique to a physical board side are entered on the individual keepout logical layer for that board side. Plots of a keepout layer for a board side are then obtained by combining the keepout master and the keepout layer for the individual board side. Adding the keepout master to the process file:

Logical Layer	Label	Description
4	KEEPOUT	Keepout master for keepout areas common to multiple layers.

If the inner layer does not contain a plane or has no need of keepout areas, then no entries are made into the keepout logical layer for that board side and no plot need be made of the keepout logical layer.

Expanding the Simple Process File to Handle More Board Sides

The simple process file described in the sections above may be expanded to accommodate many more board sides. Using the logical layer definitions and layout concepts discussed above, additional board sides may be added by defining a positive logical layer and a keepout logical layer for each additional board side. For example, suppose you wish to lay out a six sided P. C. board. The process file above could be expanded to include the following logical layers:

Logical Layer	Label	Description
13	INLAYRC	This layer contains components that represent traces, pads, and other components that represent metalization on the inner layer, C.
17	INKOC	Keepout areas for inner physical layer C.
14	INLAYRD	This layer contains components that represent traces, pads, and other components that represent metalization on the inner layer, D.
18	INKOD	Keepout area for inner physical layer D.

If you desire to keep power and ground plane information separate from the information in the other logical layers, separate logic layers may be defined for them. These logical layers could then be combined with the logical layers representing the traces, pads, and other components on a board side via the VPLOT program. To incorporate these logical layers, the process file above could be expanded to include:

Logical Layer	Label	Description
7	FIVEVOLT	This logical layer contains the pads and traces for an inner layer power plane.
9	FIVEKO	This logical layer contains the special keepout areas for an inner layer power plane.
8	GROUND	This logical layer contains the pads and traces for an inner layer ground plane.
10	GNDKO	This logical layer contains the special keepout areas for an inner layer ground plane.

Optimizing the Process File for an Output Device

The P. C. module allows for graphic output to three types of devices: the graphic CRT (display), pen plotters (pen), and photo plotters (foto). In many cases, you can build library parts more efficiently if you know which output devices are to be used. Thus, by creating multiple logical layers for the same function, you can get optimal results for each device. The supplied process file, PC.PRO, has three padmaster layers and three solder resist master layers. A good example of how these multiple layers can be used is the library part LP40O. A detailed description of LP40O can be found in the Library Parts section of this manual.

Logical Layer	Label	Description
3	PENPADM	This is the pad master layer used when doing artwork with fill to a pen plotter.
20	PENSLDM	This is the solder resist master keepout layer when doing artwork with fill to a pen plotter.
29	DISPPADM	This is the pad master layer when using the graphic CRT (during editing).
30	DISPSLDM	This is the solder resist master keepout layer when using the graphic CRT (during editing).

The process file can be expanded to maximum of 255 logical layers, thus allowing you to create separate logical layers for many different functions and components. Individual logical layers may be combined on the finished plot.

PC.PRO

The process file PC is supplied with the P. C. Board Layout module. This process file has the same functional definition of the individual logical layers as in the process file above. A listing of the process file is shown below and is followed by a functional description of the individual logical layers. Note the use of colors in the process file definition. Layers performing similar functions are assigned the same display color (for example, all keepout logical layers are displayed with the color, Aqua). In this fashion you can quickly identify and separate the information in your board design as it is displayed.

Resolution: 1 MIL is equal to 1 system grid point.

Process file is EWPC:PC

LAYER #	PEN #	LABEL	COLOR	LINE TYPE	LAYER TYPE	IDENTIFY NAME disp	IDENTIFY NAME size	TRACE NAME disp	TRACE NAME size
0	1	INSTANCE	White	Dotted	Boundary	OFF	1	OFF	1
1	1	PDRILL	Blue	Solid	Detail	OFF	1	OFF	1
2	1	BDBLANK	Yellow	Solid	Detail	OFF	1	OFF	1
3	4	PENPADM	Purple	Solid	Detail	OFF	1	OFF	1
4	1	KEEPOUT	Aqua	Solid	Detail	OFF	1	OFF	1
5	2	COMPONET	Red	Solid	Detail	OFF	1	OFF	1
6	3	CIRCUIT	Green	Solid	Detail	OFF	1	OFF	1
7	1	FIVEVOLT	Purple	Solid	Detail	OFF	1	OFF	1
8	1	GROUND	Purple	Solid	Detail	OFF	1	OFF	1
9	1	FIVEKO	Aqua	Solid	Detail	OFF	1	OFF	1
10	1	GNDKO	Aqua	Solid	Detail	OFF	1	OFF	1
11	1	INLAYRA	Purple	Solid	Detail	OFF	1	OFF	1
12	1	INLAYRB	Purple	Solid	Detail	OFF	1	OFF	1
13	1	INLAYRC	Purple	Solid	Detail	OFF	1	OFF	1
14	1	INLAYRD	Purple	Solid	Detail	OFF	1	OFF	1
15	1	INKOA	Aqua	Solid	Detail	OFF	1	OFF	1
16	1	INKOB	Aqua	Solid	Detail	OFF	1	OFF	1
17	1	INKOC	Aqua	Solid	Detail	OFF	1	OFF	1
18	1	INKOD	Aqua	Solid	Detail	OFF	1	OFF	1
19	1	REFRENCE	Yellow	Solid	Detail	OFF	1	OFF	1
20	1	PENSLDM	Aqua	Solid	Detail	OFF	1	OFF	1
21	1	SLDRCOMP	Aqua	Solid	Detail	OFF	1	OFF	1
22	1	SLDRCIRC	Aqua	Solid	Detail	OFF	1	OFF	1
23	1	UPDRILL	Blue	Solid	Detail	OFF	1	OFF	1
24	1	CIRCKO	Aqua	Solid	Detail	OFF	1	OFF	1
25	1	COMPKO	Aqua	Solid	Detail	OFF	1	OFF	1
26	1	ASMBDRAW	Yellow	Solid	Detail	OFF	1	OFF	1
27	4	FOTOPADM	Purple	Solid	Detail	OFF	1	OFF	1
28	1	FOTOSLDM	Aqua	Solid	Detail	OFF	1	OFF	1
29	4	DISPPADM	Purple	Solid	Detail	OFF	1	OFF	1
30	1	DISPSLDM	Aqua	Solid	Detail	OFF	1	OFF	1

Unused memory = 1410 Kbytes

Logical Layer	Label	Functional Description of Contents
0	INSTANCE	This layer is reserved by the system to display instances of library parts.
1	PDRILL	This layer contains the component with the iname ORIGIN, and all circles representing plated holes.
2	BDBLANK	This logical layer contains components that represent the P. C. board outline and the tooling hole pads for the router.
3	PENPADM	This logical layer contains components that represent areas that are to become copper on both the component and circuit sides of the board for pen plots with fill.
4	KEEPOUT	This logical layer is the keepout master. It contains components that represent all keepout areas common to multiple layers.
5	COMPONET	This logical layer contains components that represent traces, pads, and other components representing metalization on the component side of the board.
6	CIRCUIT	This logical layer contains components that represent traces, pads and other components representing metalization on the circuit side of the board.
7	FIVEVOLT	This logical layer contains the pads and traces for an inner layer power plane.
8	GROUND	This logical layer contains the pads and traces for an inner layer ground plane.
9	FIVEKO	This logical layer contains the special keepout areas for an inner layer power plane.
10	GNDKO	This logical layer contains the special keepout areas for an inner layer ground plane.
11	INLAYRA	This logical layer contains components that represent traces, pads, and other metalization on the inner layer, A.
12	INLAYRB	This logical layer contains components that represent traces, pads, and other metalization on the inner layer, B.
13	INLAYRC	This logical layer contains components that represent traces, pads, and other metalization on the inner layer, C.
14	INLAYRD	This logical layer contains components that represent traces, pads, and other metalization on the inner layer, D.
15	INKOA	This logical layer contains the special keepout areas for the inner layer, A.
16	INKOB	This logical layer contains the special keepout areas for the inner layer, B.
17	INKOC	This logical layer contains the special keepout areas for the inner layer, C.
18	INKOD	This logical layer contains the special keepout areas for the inner layer, D.
19	REFERENCE	This logical layer contains the registration graphics and text for the P. C. fabrication shop.
20	PENSLDM	This logical layer contains components that represent areas on both outer layers of the board that are to be void of solder resist for pen plots with fill.
21	SLDRCOMP	This logical layer contains components that represent areas unique to the component side of the board that are to be void of solder resist.

Logical Layer	Label	Functional Description of Contents
22	SLDRCIRC	This logical layer contains components that represent areas unique to the circuit side of the board that are to be void of solder resist.
23	UPDRILL	This logical layer contains the component with the iname, ORIGIN. It contains all circles representing unplated through holes.
24	CIRCKO	This logical layer contains components which represent the special keep-out areas for the circuit side of the board.
25	COMPKO	This logical layer contains components which represent the special keep-out areas for the component side of the board.
26	ASMBDRAW	This logical layer contains the outlines of parts for assembly drawings.
27	FOTOPADM	This logical layer contains components that represent areas that are to become copper on both the component and circuit sides of the board for photo plotting.
28	FOTOSLDM	This logical layer contains components that represent areas on both outer layers of the board that are to be void of solder resist for photo plotting.
29	DISPPADM	This logical layer contains components that represent areas that are to become copper on both the component and circuit sides of the board for the display.
30	DISPSLDM	This logical layer contains components that represent areas on both outer layers of the board that are to be void of solder resist for the display.

This process file is sufficient for laying out six layer P. C. boards, if you adhere to the layout concepts described in the sections above. Of course, you may modify this process file or scrap it altogether if it does not meet the needs of your fabrication process.

One significant aspect of the process file, PC.PRO, is that it is the process file used to create the library of parts provided with this module. This means that the individual components that form a device are present on a specific layer to perform a specific function when output. For example, the library part, PAD25S, consists of 50x50 mil pads on logical layers 3, 27 and 29 (pad masters for pen plotting, photo plotting and display), 80x80 mil areas on logical layers 20, 28 and 30 (solder resist keep outmasters for pen plotting, photo plotting and display), a circle on layer 1 with the iname '.00025' specifying the drill hole, two lines on layer 1 that cross over the drill hole (for drill drawings) and a 100x100 mil area on layer 4 (keepout master). Adding an instance of the library part, PAD25S, causes the component parts to be added to the predefined layer number (i.e., pad on logical layer four, pad on logical layer three, and circle on layer one), regardless of the process file definitions. Therefore, if you intend to modify PC.PRO and still use the supplied library parts, you need to edit and modify those devices so that component parts are displayed and plotted with the correct logical layer (as defined in your modified process file).

The Macro File

PC.MAC is a macro file supplied with the P. C. Board Layout Module in the volume EWPC. Most macros in the macro file are copies of the ones in the General Drawing module's macro file, and thus are not discussed here. The new macros are defined below:

DISPDISP	Executes the proper SHOW command to turn on the display specific layers while turning off the pen plot and photo plot layers.
DISPPEN	Executes the proper SHOW command to turn on the pen plot specific layers while turning off the display and photo plot layers.
DISPFOTO	Executes the proper SHOW command to turn on the photo plot specific layers while turning off the display and pen plot layers.

Supplied Library Parts

A set of library parts is provided with the P. C. Layout module in the EW volume, EWPC. As mentioned above, these library parts conform to the process file, PC.PRO, provided with this system. Each library part consists of components (lines, rectangles, circles, instances of other library parts, etc.) added to specific logical layers.

Because each P. C. fabrication shop has its own set of specifications for P. C. board layout (i.e., tolerances, pad sizes, drilled hole sizes, photo plotter aperture sizes, etc.), you will want to create your own set of library parts. Therefore, those library parts supplied with this module form a reference for constructing your own set of library parts. The library parts supplied take advantage of certain techniques that utilize the capabilities of the photo plotter and of the photo plotter programs. These techniques are highlighted when discussing the individual library parts; however, detailed discussions are deferred until the sections of this manual dealing with the creation of virtual plot files and creation of photoplotter data.

The set of library parts can be broken up into seven basic categories: reference parts (such as registration marks, sheer marks, etc.), primitive pads, primary pads, inner layer pads, IC DIPs (Integrated Circuit Dual In-line Packages), log parts, and edge connectors. Each category of parts is discussed in the following sections. A pictorial representation of each supplied library part and a listing of the Graphics Editor ADD commands used to create each part (a copy of the ARCHIVE file), is provided at the end of this chapter.

Library parts are often created using other library parts as a foundation. For example, PAD25S actually consists of an instance of the library part LP25S and a 100x100 mil area on logical layer 4 (the keepout master). This means that due care must be exercised when modifying a library part, as not only the specific device is modified, but any device linked to the device is modified as well.

Reference Library Parts

Reference library parts are components added to a P. C. board layout that are used to align the physical board layers. The reference parts supplied are: a tool hole, a shear mark (also called a corner), and a registration target. Each of these parts are described below.

The Tool Hole

The tool hole library part TOOLHOLE is designed to provide reference points to help locate and position the board for machining. The origin of this library part is at the center of the pads and drilled hole. The iname assigned to the circle in layer 3 tells the photoplotter programs to flash this component with a special aperture instead of tracing it.

The Shear Mark

The shear mark library part CORNER is required at each corner of the P. C. board to bracket the board outline for the routing process. Since only one orientation of the device CORNER is supplied, you will need to use the mirror and rotate functions of the ADD command to obtain the other three orientations which you may require. The origin of the library part CORNER is located at the intersection of the two line segments that form the part.

The Target

The library part TARGET is the target registration mark used to align different plots that comprise a physical board layer. It is also the mark used to align the physical layers to form a board. This library part consists of concentric diamonds made of 12 mil wide lines, two perpendicular 12 mil wide lines that form a cross-hair, and a small rectangle. All of the 12 mil wide lines that form the library part have the iname, ZNULL, which tells the photo plotter programs not to photo plot these components. The small rectangle has the iname, ZTARGET, which tells the photo plotter programs to substitute and flash a special target aperture (usually a moire) instead of the rectangle. The origin of this library part is the center of the concentric diamonds.

Primitive Pads

Primitive pads represent outer layer pads on the master layers, for copper and solder resist keepout areas and a drilled hole in your drawing. Each primitive pad library part is named according to the finished hole size (FHS) and the shape of the pad. The two leading characters of the name (LP) in each primitive pad name identify the library part as a primitive pad. The next two characters in the library part name are digits that represent the finished hole size, in mils. The final character in the library part name is a letter specifying the shape of the pad: (C) for circular pads, (R) for rectangular, (S) for square, and (O) for oval.

Each primitive pad consists of an outer layer pad shape on logical layers 3, 27 and 29 (the pad master logical layers), a solder resist keepout shape on logical layers 20, 28 and 30 (the solder resist keepout logical layers) and a circle on logical layer 1 the PDRILL information logical layer) with the iname .Dxxxx (where xxxx represents the finished hole size in miles). The origin of each primitive pad library part is the center of the pad, which is also the center of the drill hole.

The pad for the library part LP40O appears as a polygon in the shape of an oval on the display layer (layer 29), a line with width wrapped in on itself (to allow filling) in the shape of an oval on the pen plot layer (layer 3) and a rectangle with the iname .ZOVAL (to force flashing) on the photo plot layer (layer 27).

The solder resist keepout for the library part LP400 appears as a polygon in the shape of an oval on the display layer (layer 30), a line with width wrapped in on itself (to allow filling) in the shape of an oval on the pen plot layer (layer 20) and a short 80 mil wide line on the photo plot layer (layer 28). Remember that the photo plotter draws lines by moving a round trace aperture from one end point to another. In the case of an 80 mil wide line, the 80 mil wide aperture will be selected, and the resulting figure on the film will have nicely rounded endpoints, thus we get an oval. This demonstrates a technique that allows the photo plotter to quickly plot this point.

The pad for the library part LP40O consists of rectangles, the display layer (layer 29), and pen plot layer (layer 3), and as two overlapping squares on the photo plot layer (layer 27). If you look at wheel1 you will see that these squares can be flashed. This is another technique you can use to speed up photoplotting.

Primary Pads

Primary pad library parts are groups of components that represent the components required for a plated through hole on a P. C. board. Primary pads are named in the same fashion as primitive pads: two digits in the library part name specify the finished hole size of the library part, and a trailing character, (C for circular, O for oval, S for square, or R for rectangular), that identifies the shape of the outer layer pad.

Each primary pad library part consists of: a square in layer four (the KEEPOUT master logical layer), and an instance of a primitive pad library part. Thus, if the techniques and concepts described with the process file are adhered to, all of the components necessary for creating a plated through hole are present in each primary pad library part.

Inner Layer Pads

Inner layer pads are devices composed of components that represent pads on inner layers which connect to plated through holes. The names of the inner layer pad library parts begin with the two characters, IP, identifying it as an inner layer pad. The next two characters of the library part name are digits representing the diameter of the inner layer pad. Pads with a diameter of 85 mils are designed for connecting to plated through holes with a finished hole size of 40 or 46 mils. Pads with a diameter of 70 mils are designed for connecting to plated through holes with a finished hole size of 25 mils. The remaining characters in the library part name identify the logical layer in which the inner layer pad is to be added. For example, the inner layer pad library part, IP85C, consists of an 85 mil diameter pad in logical layer 13, the logical layer reserved for positive components on inner layer, C.

Two inner layer pad library parts, IP85FIVE and IP85GND, also provide a twenty mil wide line in the appropriate inner positive layer that dissects the inner layer pad and extends well beyond the boundary of the pad. These lines connect the inner layer pad to traces and power and ground planes located in the same layer as the inner layer pad.

I.C. DIPs

I.C. DIP (Integrated Circuit Dual In-line Packages) library parts are composed of components representing the pads, drilled holes, keepout areas, etc., and an I.C. DIP. Each I.C. DIP library part consists of: two 120 mil wide rectangles in logical layer four (the keepout master), an assembly outline in layer 26, and two columns of primitive pads that provide the drill information and the outer layer pad to the pad master. The I.C. DIP library parts use instances of the primitive pad LP40O to provide oval pads for all of the DIP pins, except for pin number one. An instance of LP40R is used to provide the pad for pin number one, because it provides a rectangular pad instead of an oval pad. This makes pin number one visually easier to identify on the photo plotted artwork.

The dimensions of each DIP library part are the same (i.e., the physical spacing between adjacent pins of a dip are the same, and the physical distance between rows of pins are the same), only the number of pins change. Drilled holes for each pin of a DIP library part are 40 mil drilled holes that are represented by circles in logical layer one.

Two large rectangles (one per column of pins) added to logical layer four (the keepout master) insure that traces don't unintentionally short the DIP pins. Instances of the primitive pads, LP40O and LP40R, provide the pads on the outer board layers (via the pad master logical layers) and the drill hole information (vis circles added to logical layer one).

The photo plotter draws and fills lines by selecting a circular aperture that matches the width of the line to be drawn. Then by shining light through the aperture, it moves the center of the circular aperture from one endpoint of the line to the other. Because a circular aperture is used, the end points of the line are rounded and extend beyond the end points of the line by a distance equal to the radius of the aperture used to draw and fill the line. Thus by using an 80 mil wide by 40 mil long line in logical layer 28, an oval-like keepout shape is generated on the photo plotter for the solder resist keepout master. This concept of rounding due to circular apertures is discussed in greater detail in the sections of this chapter dealing with the photo plotter programs.

Edge Connectors

The edge connector library parts are composed of components that represent connector fingers and the board blank outlines of an edge connector. Each edge connector library part represents a portion of a final edge connector. These portions of edge connectors can be combined in many different permutations to provide the edge connector that you need in a particular application.

Edge connector fingers consist of a connector pad on the pad master logical layers, and a line representing the edge of the board on logical layer two (the board blank logical layer). No keepout regions are provided for the solder resist logical layer. The origin of the connector finger is at the intersection of the centerline of the pad and the board edge. The finger is oriented for a connector along the bottom edge of the board; however, by utilizing the rotate and mirror functions of the ADD command, any orientation may be obtained.

Two different connector fingers are provided in the library: EC100F and EC156F. These two fingers are provided to allow for different finger size and finger spacing requirements of an edge connector.

EC100F is for connectors with a 100 mil spacing and consists of a 50 mil by 250 mil pad in the pad master logical layers and a line of zero width in logical layer two (the board blank logical layer). It is designed such that when photo plotted, a 30 mil clearance is created between the end of the finger and the edge of the board (due to rounding by the photo plotter when drawing lines with circular apertures). However, when plotted on an HP pen plotter, a clearance of 55 mils is obtained between the end of the finger and the edge of the board.

EC156F is for connectors with a 156 mil spacing and consists of an 80 mil by 300 mil pad in the pad master logical layers and a line of zero width in logical layer two (the board blank logical layer). Like EC100, this library part is designed to provide a 30 mil clearance between the end of the connector fingers and the end of the board due to rounding by the photo plotter when plotting lines with a circular aperture). When EC156F is plotted on an HP pen plotter, a clearance of 70 mils is obtained between the end of the finger and the edge of the board.

Board blank outline parts, EC100BB and EC156BB, consist of lines on logical layer two (the board blank logical layer) that form the outline of the edge connector. Two different sizes of board blank outlines are provided: one for edge connectors with 156 mil spacing and one for edge connectors with 100 mil spacing. Each part is oriented for an edge connector along the bottom edge of the board. Different orientations may be obtained by utilizing the mirror and rotate functions of the ADD command.

Producing Punch Tape for N/C Drill Machines

The P. C. Board Layout module provides the capability to produce punch tapes on the HP 9884 tape punch for Excellon format N/C drill machines. Later sections of this chapter deal with modifying the module to allow other tape punches (or N/C devices) and/or non-Excellon N/C formats to be used. Punched tapes are produced from a generate file by a four step process. First, a P. C. board layout drawing is created with the commands of the edit subsystem. A generate file is created from your drawing. A virtual drill file is created (by a supplied program, named VDRILL) from the generate file. And finally, a punched paper tape is created on an HP 9884 from the virtual drill file. You may optionally store this N/C drill information in a BASIC data file.

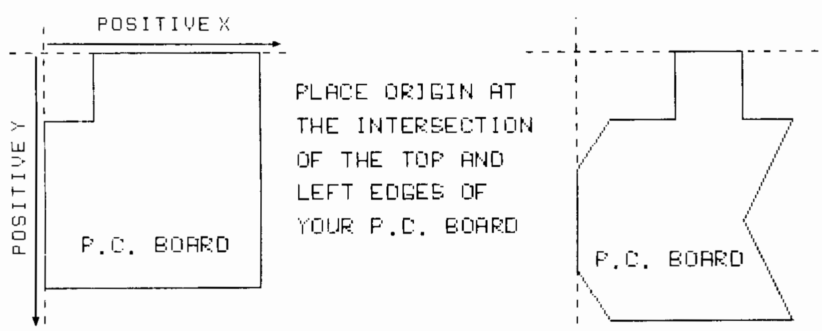
Many drawings can be combined when creating a virtual punch file. This allows you to segment your design over many drawings. These drawings can then be combined in the virtual drill file. When a virtual drill file is converted into a punched tape, a BASIC program (called PDRILL) reads the virtual drill file. It then massages the data into Excellon format, and outputs the final drill data to an HP 9884 tape punch or to a BASIC data file. For the supplied N/C drill programs to work correctly, several conventions must be followed when laying out your P. C. board.

Conventions

For the system to produce N/C drill information from drawings, there must be some way to recognize components representing drill holes and there must be some method of determining the size of the hole to be drilled. Therefore, a set of conventions for P. C. board layout have been defined. These conventions must be followed if you desire to use the supplied programs to produce N/C drill information.

Only circles with a special iname are considered to be drilled holes. The iname that you assign to the circle component tells the VDRILL program (the program creating the virtual drill file) what size you desire the drilled hole to be. The iname must be of the form Dxxxx, where the letter D identifies the circle as a drill hole and the trailing characters xxxx specifies the drilled hole size, in mils. For example, the iname D0050 specifies a hole size of 50 mils while the iname D00500 specifies a hole size of 500 mils. A report generated by the PDRILL program then assigns a specific N/C tool number to each different hole size specified with the iname.

The VDRILL program requires that a reference circle with the iname ORIGIN be added to each layer containing drill information. The reference circle must be placed at the intersection of the top edge and the left edge of the board. The reference circle may be of any size or resolution. The VDRILL program then uses this reference circle as a zero reference and recalculates the position of all drill holes relative to the position of the reference. This causes VDRILL to produce a zero offset drill tape.



The VDRILL and PDRILL programs cannot produce N/C drill information from a drawing stored in a device file. Drawings must first be operated on with the GENERATE command in the Graphics Editor to produce a generate file. The generate file is then operated on by the VDRILL program to produce a virtual drill tape output. This is then stored in a virtual drill file which is later passed to the PDRILL program.

In summary, the following list of conventions must be used when laying out your P. C. board if you intend to use the VDRILL and VDRILL programs:

- Holes to be drilled must be represented by circle components.
- Circles representing holes to be drilled must have an iname of the form, Dxxxxx, where xxxxx represents the finished size of the drilled hole, in mils.
- A circle with an iname, ORIGIN, must be placed at the intersection of the top and left edge of the board.
- A generate file must be created from the layout drawing. The generate file(s) is then operated on by the VDRILL program and passed to the PDRILL program to produce the finished N/C data.

The VDRILL Program

The VDRILL program is an EW program that combines one or more generate files into a virtual drill file. The PDRILL program then uses the virtual drill file to produce Excellon format N/C data. The VDRILL program is accessed by selecting Create Virtual N/C Drill File from the Manager menu.

Once the program is selected, the system prompts you for the name of the generate file created from your drawing. You may respond with a carriage return (press CONTINUE) to return to the Manager menu. Responding with a file specifier identifies the generate file upon which the program is to act. A file specifier consists of an optional volume name and colon followed by the name of the file. You need to specify a volume name only if it is different than the system volume. For example, EWPC:EXAMP is a file specifier specifying a file named EXAMP that is located on the volume EWPC. No file suffix is required since the system knows that only generate files may be operated on by VDRILL.

Next the system prompts you for a file specifier defining the name that you want assigned to the virtual drill file and the EW volume on which the file is to reside. If you do not specify a file name in response to the prompt, VDRILL assigns the virtual drill file the same file name as the first generate file used to create it (however; it adds the suffix, .VDR when creating the file on the specified volume).

The system next prompts you to enter the layer numbers of the file which contain drill information. The information is entered by typing each layer number followed by pressing CONTINUE. To specify no layers, simply press CONTINUE. To specify all layers, enter A CONTINUE.

Next the system prompts you for the major axis of your board. If most of the drilled holes are aligned along the Y-axis (vertically), enter Y CONTINUE. If most of the drilled holes are aligned along the X-axis (horizontally), enter X CONTINUE. If neither axis prevails, use either X or Y.

Once the VDRILL program reads the data from the EW file, it prompts you for additional generate files. Responding with a carriage return (no file name is supplied, only CONTINUE is pressed) causes VDRILL to operate on the file(s) supplied. The Manager menu is then displayed. Responding with another file specifier causes the system to assume that you wish to combine the drill information in the new file with that in the original file. The system continues to prompt you for additional files until you respond with only a carriage return. Because of this feature, you may segment your board design, storing parts of board design in different drawings and thus reducing congestion on the CRT in complex drawings. These drawings may then be recombined using the supplied N/C drill and plot programs.

When you have finished supplying generate file names to the VDRILL program, it produces the virtual drill file.

The Virtual Drill File

The virtual drill file is an EW file that contains generalized information for N/C drills. A listing of a sample virtual drill file is provided:

```
40              (Finished hole size)
10,72          (x,y coordinates)
10,105
10,693
1040,6222
5000,5000
##EOP##        (End of drill path marker)
55             (Finished hole size)
15,72          (x,y coordinates)
15,105
##EOP##        (End of drill path marker)
119           (Finished hole size)
85,85          (x,y coordinates)
100,100
##EOT##        (End of tape marker)
```

The PDRILL Program

The PDRILL program is a BASIC program that reads a virtual drill file, converts it to Excellon format, and outputs the converted file. The PDRILL program is accessed by selecting Create N/C Drill File from the Manager menu.

Once selected, the system prompts you to enter a file specifier identifying the file into which to place the drill information. Entering `PUNCH:` causes the drill information to be sent to the HP 9884. Any other entry causes the output to be placed in a BASIC data file of that name. Entering `TAPE:` causes the drill information to be sent to the HP 7970E mag tape drive.

Next the system prompts you to enter a file specifyer identifying the virtual drill file upon which to operate.

Once the virtual file name is specified, the system reads in the file, converts it to Excellon format with locations specified in ASCII code, and outputs the finished file to the specified location. If used, the system must know where to find the tape punch (The HP 9884 must be defined in the system configuration) before attempting to run PDRILL. Therefore, the HP 9884 Tape Punch must be added to the system configuration via the System Configuration module. For the PDRILL program to function properly, the HP 9884 should be added to slot number nine (#9) in the peripheral table. If used, the HP 7970E mag tape drive must be added to slot ten in the peripheral table.

Once the output is produced the system produces a report on the system printer that tells you the number of holes of each size that appear on the board. It also tells you the N/C tool number assigned to each hole size. This tells you what size drill is to be placed in each tool location of your N/C drill machine. An example of the report is shown:

```
Tool  1 -- Hole Size = 40                    Number of holes =  22
Tool  2 -- Hole Size = 119                   Number of holes =  3

Total Holes =  25
```

The PDRILL program produces punch tapes in the following format:

```
T01
XnnnnnYnnnnn
XnnnnnYnnnnn
T02
XnnnnnYnnnnn
M30
```

Where:

TXX	-identifies which tool number is to be used to drill the hole.
nnnnn	-is a five digit field (padded to the left with zeros) that specifies the location relative to the component ORIGIN where the hole is to be drilled.
M30	-specfies the end of the tape and is followed by a carriage return and a line feed character.

Each line of the drill output is terminated with an ASCII carraige return/line feed character.

PDRILL produces a series of one hundred NULL characters as a punch tape header (mag tape is unlabeled) and a tape trailer that consists of the characters, M30, and a series of one hundred NULL characters.

Modifying the Drill Code Format and Changing the Output Device

Because N/C codes are not standard throughout the industry, you may need to change the format of the N/C tape produced or specify an output device other than those supported. Since the PDRILL program is written in the BASIC programming language, you may modify the program to meet the needs of your fabrication shop. The program has been written in a modular fashion (using subprograms to segment the tasks) and fully commented to help you understand what areas you need to modify. Additionally, the following list of subprograms and their functions should help in directing you to the area to be modified. A thorough understanding of the BASIC programming language, and in particular, of subprograms, parameter passing, and I/O statements is necessary before attempting to modify the program.

1. Subprogram Header- This subprogram produces a series of one hundred NULL characters at the beginning of the tape.
2. Subprogram Tool_change- This subprogram is responsible for writing the tool change commands (i.e., T01, T02, etc.).
3. Subprogram Trailer- This subprogram produces the characters M30 and a series of one hundred NULL characters at the end of the tape.
4. Subprogram Pad_integer- This subprogram is responsible for padding integers to a given length by adding leading zeros.
5. Subprogram COORD- outputs x and y coordinate data.
6. Subprogram Output- This subprogram is responsible for sending the formatted output to the HP 9884 Tape Punch, the HP 7970E mag tape drive, or BASIC data file.

You are free to direct output to any device that is available to the BASIC language (any device supported by the Mass storage ROM or the I/O ROM). To do this you must modify or rewrite the Output subprogram in the PDRILL program. By simply writing a new output routine (in BASIC or ASSEMBLY language) you may direct the output to any device that is supported by your computer.

The PDRILL program outputs N/C drill code information using seven bit ASCII code. Since some N/C drill machines require an EIA code instead, a software switch is provided that allows you to specify output in EIA code instead of ASCII. This software switch is the variable, Output_eia in the Output subprogram. Assigning this variable a non-zero value causes the system to output EIA code instead of ASCII.

Photo Plotting

The P. C. Board Layout module provides two programs, VPLOT and PPLOT, that allow you to produce data to drive a Gerber photo plotter. These programs are accessed by selecting Create Virtual Plot File and Create Photoplotter Mag Tape, respectively. We will use the program names VPLOT and PPLOT throughout the discussion of photo plot generation. The supplied programs allow you to produce plots of individual logical layers or combinations of logical layers. Before discussing the individual programs, it is necessary to have a basic understanding of how a photo plotter creates a plot and the peculiarities of working with such a device.

The Photo Plotter

Photo plotters produce plots on light sensitive film by shining light onto the film through specially shaped holes, called apertures. By moving the aperture over the film while shining light through it, it is possible to draw long lines and large shapes. Gerber photo plotters are capable of holding 24 apertures in a specially designed wheel called an aperture wheel. When photo plotting, the photo plotter spins the wheel to position the correct aperture in front of the light and then begins plotting. Apertures come in a variety of shapes (circles, ovals, squares, etc) and are equipped with one of two different filters which allow different amounts of light to reach the film. These filters are used for creating plots in one of two ways: flashing and tracing.

When a component is plotted by tracing, the photo plotter selects an aperture (or an aperture specified by you) and traces the figure by shining light through the aperture as it moves. If the figure is to be filled, the photo plotter fills the figure by continually tracing the figure and overlapping the traces until all of the figure is filled.

While tracing provides the desired output, it is a time-consuming process. Another method of photo plotting, called flashing, involves the use of specially shaped apertures or apertures of a particular size and dimension. When a component is plotted by flashing, the photo plotter selects an aperture of the same shape and dimension of the component to be plotted and flashes light through the aperture, producing a copy of the aperture on the plot. This is clearly a more time efficient method of producing filled plots than tracing. Special figures (such as registration targets, ovals, etc.) that are difficult to trace can be added to the Gerber wheel to be flashed. A disadvantage of flashing is that you must have apertures the exact size and shape of the component to be plotted.

A normal configuration of a photo plotter wheel includes apertures for tracing and for flashing. The VPLOT and PPLOT programs take full advantage of both capabilities.

Conventions

If you intend on using the photo plotter programs VPLOT and PPLOT that are supplied with this application module, certain conventions of board layout and file structure need to be followed. These conventions are detailed in the sections that follow.

For the photo plotter programs to produce a photo plot, they must have a method of knowing what apertures are available on the aperture wheel of the photo plotter. This information is supplied to the program through a table, called a Gerber Aperture Table. The Gerber Aperture Table is stored in a BASIC data file called a wheel file. You will create many wheel files since each file is different for each set of apertures used. Then using the File Utilities Module, you must transfer this information into an EW file, allowing the EW program VPLOT to access it. The Gerber Aperture Table consists of five columns of information for each aperture on the wheel. This required information is described below. It is followed by a description of how to construct your wheel file and a listing of a sample wheel file. A sample wheel file is supplied with your system and is stored in the BASIC environment with the name, WHEEL1. This same file has been transferred into the EW volume EWPC, and is stored there with the name, WHEEL1.

```
10 ! W10 10 Circle Trace 10.
20 ! W12 11 Circle Trace 12.
30 ! W15 12 Circle Trace 15.
40 ! W20 13 Circle Trace 20.
50 ! W25 14 Circle Trace 25.
60 ! W50 15 Circle Trace 50.
70 ! W60 16 Circle Trace 60.
80 ! W80 17 Circle Trace 80.
90 ! W140 24 Circle Trace 140.
100 ! R55 28 Circle Flash 55.
110 ! R65 19 Circle Flash 65.
120 ! R70 70 Circle Flash 70.
130 ! R80 20 Circle Flash 80.
140 ! R90 21 Circle Flash 90.
150 ! R100 22 Circle Flash 100.
160 ! R150 25 Circle Flash 150.
170 ! S50 27 Square Flash 50.
180 ! S55 18 Square Flash 55.
190 ! S70 71 Square Flash 70.
200 ! S100 23 Square Flash 100.
210 ! OVAL 29 Rect Flash 75. 55.
220 ! OVAL 72 Rect Flash 55. 75.
230 ! DIAM 26 Square Flash 0.
240 ! TARGET 73 Square Flash 0.
```

1. The aperture name - an alphanumeric name, six characters maximum, that you assign to the aperture so that you may associate it with a particular component in your board layout.
2. The Gerber select code associated with the individual aperture. Each individual aperture on a Gerber aperture wheel is numbered from one to twenty four. However, the photo plotter requires that specific Gerber select codes be assigned to the individual apertures. The table immediately preceding the sample wheel file shows the required aperture assignments.
3. The shape identifier - specifies the physical shape of the aperture. Three shape identifiers are recognized by VPLOT: Circle, Square, and Rect. For asymetrical shapes (i.e., those whose length dimension is not equal to its width dimensions; such as ovals, rectangles, etc.) the shape identifier, Rect, may be used.

VPLOT uses this information to decide whether it can flash a component as opposed to tracing it. VPLOT attempts to plot every rectangle or circle component with an iname of the form, Zxxxxx with a flash aperture having the name xxxxx in the wheel file. If the aperture name exists in the file, the component is flashed with the aperture. However, if the shape identifier is Rect, then the aperture is flashed only if the component is a rectangle component and the dimensions of the component match (with the tolerance specified with the :E parameter and the VPLOT) those of the aperture.

If the :F parameter is specified with the VPLOT command, VPLOT tries to match the dimensions of each component to the dimensions of each aperture. If a match is found, VPLOT then checks to see what shape identifier is assigned to the aperture. If the shape identifier is Circle or Square, the figure is flashed. This allows you an extra measure of control in determining whether a particular aperture is flashed or not.

4. The aperture filter identifier - specifies whether the aperture is equipped with a filter for tracing or for flashing. Only two filter identifiers are recognized: Trace and Flash.
5. The aperture dimension - specifies the actual physical dimension of the aperture. For circular and square apertures, this value represents the diameter and length, respectively. For asymetrical shapes (such as ovals, rectangles, etc.) two values must be supplied, the length and width (in that order) which represent the major and minor axes of the aperture. For special apertures that you don't want matched (VPLOT considers all flashable apertures fair game in replacing components with flashes) specify a zero (0.) dimension. Be sure to include the trailing decimal point.

Table of Gerber Select Codes and Aperture Positions

Aperture Wheel Position	Gerber Select Code
1	10
2	11
3	12
4	13
5	14
6	15
7	16
8	17
9	18
10	19
11	70
12	71
13	20
14	21
15	22
16	23
17	24
18	25
19	26
20	27
21	28
22	29
23	72
24	73

The wheel file that you must create is nothing more than a BASIC language program made up of all comment statements. Each comment line of the program must contain each of the five values listed above and thus completely describe a single aperture. A maximum of 200 apertures may be defined per wheel file. This allows non-Gerber photo plotters to be utilized with the system (Gerber photo plotters use a maximum of twenty-four apertures per wheel). Apertures must be arranged in the table in ascending order according to the physical size of the aperture. Trace apertures must be grouped together and arranged in ascending order at the beginning of the wheel file. Apertures with the same name but rotated at an angle of 90 degrees, must be adjacent in the wheel file. For example, see the two entries for OVAL in the supplied wheel file, WHEEL1.

You may assign any name you wish to the BASIC language wheel file. However the wheel file must be placed on the mass storage medium using the BASIC langauge SAVE statement, not the STORE statement. Then using the File Utilities function BASIC to EWE, you must tranfer the BASIC file into an EW volume. This allows the VPLOT program to recognize and access the wheel file.

If you desire that a particular component be photo plotted by flashing with a particular aperture, you must assign a special iname to the component. Using an iname of the form Zxxxxxx causes the VPLOT program to attempt to photo plot the component by flashing. The characters xxxxxx should be the aperture name appearing in the Gerber Aperture Table for the the aperture with which you wish to flash. For example, assigning the iname ZTARGET to a component causes the VPLOT program to check the Gerber Aperture Table to see if there is a flashable aperture with the name, TARGET. If VPLOT finds the entry in the table, it substitutes the flashed figure for the original component in the final photo plot.

Any component that has an iname that starts with the letter 'Z' will be considered for substitution by VPLOT. However, VPLOT allows only circles and rectangles to be replaced in this manner.

Any component with the iname ZNULL will not be photo plotted at all. This is a convenient feature for stripping out components that you want displayed on the CRT, but do not want to appear in the photo plot. The supplied library part TARGET takes advantage of this feature and provides a good example of how to use inames to their fullest potential. The use of the iname ZNULL and the use of the iname ZTARGET cause the library part, TARGET, to be totally replaced in the photo plot by a flash of the aperture named, TARGET.

Like the VDRILL program, the photo plot programs cannot produce a photo plot from a drawing stored in a device file. Drawings must first be operated on with the GENERATE command in the Graphics Editor to produce a generate file. The generate file is then operated on by the VPLOT program which passes the information on to the PPLOT program.

Creating a Virtual Plot File

Selecting Create a Virtual Plot File allows you to access the VPLOT program. The VPLOT program is an EW program that takes information stored in one or more generate files to produce a virtual plot file. A virtual plot file is a partial set of commands to reproduce the original drawing(s) on a Gerber photo plotter. The PPLOT program then operates on this virtual plot file to produce the final set of commands that drives a Gerber photo plotter.

Once this branch is selected, the system displays the prompt: -->. You may enter one of three commands in response to this prompt: BYE, HELP, or VPLOT. The syntax of these commands and a description of the function of each is provided in the syntax reference section. A short functional description is provided below.

The BYE Command

The BYE command tells the system that you do not wish to continue working with the VPLOT program and that you want to return to the P. C. Layout module menu without any further processing of the data.

The HELP Command

The HELP command causes the system to display the syntax of the BYE, HELP, and VPLOT commands.

The VPLOT Command

The VPLOT command allows you to create a virtual plot file from a generate file (created from your P. C. drawing). The PPLOT program then converts this virtual plot file into photo plotter data for a Gerber 4300/4400 series photo plotter.

The Virtual Plot File

The virtual plot file is an EW file that contains both the Graphics Editor commands used to create the drawing and the Gerber photo plotter commands necessary to produce the plot on a Gerber Photo Plotter. It is created by the VPLOT program and has the following format:

One complete ADD command from the generate file
Gerber commands necessary to plot component described by the ADD command
One complete ADD command from the generate file
Gerber commands necessary to plot component described by the ADD command

.

.

.

A listing of a sample virtual plot file is provided:

```
ADD N19 :F100. "EXAMPLE REV.A"  -1800., 1500.;
G54D11*G01X4932Y3237D02*G01X4882Y3237D01*G01X4882Y3275D01*G01X4907Y3275D01*G01X4
882Y3275D01*G01X4882Y3312D01*G01X4932Y3312D01*G01X49
57Y3312D02*G01X5007Y3237D01*G01X4982Y3275D01*G01X4957Y3237D01*G01X5007Y3312D01*G
01X5032Y3312D02*G01X5032Y3250D01*G01X5045Y3237D01*G01
X5070Y3237D01*G01X5082Y3250D01*G01X5082Y3275D01*G01X5032Y3275D01*G01X5082Y3275D0
1*G01X5082Y3312D01*G01X5107Y3312D02*G01X5107Y3237D01*
G01X5132Y3262D01*G01X5157Y3237D01*G01X5157Y3312D01*G01X5182Y3312D02*G01X5182Y323
7D01*G01X5220Y3237D01*G01X5232Y3250D01*G01X5232Y3262D
01*G01X5220Y3275D01*G01X5182Y3275D01*G01X5257Y3237D02*G01X5257Y3312D01*G01X5307Y
3312D01*G01X5382Y3237D02*G01X5332Y3237D01*G01X5332Y32
75D01*G01X5357Y3275D01*G01X5332Y3275D01*G01X5332Y3312D01*G01X5382Y3312D01*G01X54
82Y3312D02*G01X5482Y3237D01*G01X5520Y3237D01*G01X5532
Y3250D01*G01X5532Y3262D01*G01X5520Y3275D01*G01X5482Y3275D01*G01X5495Y3275D01*G01
X5532Y3312D01*G01X5607Y3237D02*G01X5557Y3237D01*G01X5
557Y3275D01*G01X5582Y3275D01*G01X5557Y3275D01*G01X5557Y3312D01*G01X5607Y3312D01*
G01X5632Y3237D02*G01X5657Y3312D01*G01X5682Y3237D01*G0
1X5726Y3318D02*G01X5726Y3306D01*G01X5738Y3306D01*G01X5738Y3318D01*G01X5726Y3318D
01*G01X5782Y3312D02*G01X5782Y3250D01*G01X5795Y3237D01
*G01X5820Y3237D01*G01X5832Y3250D01*G01X5832Y3275D01*G01X5782Y3275D01*G01X5832Y32
75D01*G01X5832Y3312D01*
ADD L5 :W15.
  -2050., -475.  -1850., -475.  -1850., -525.  -2175., -525.
  -2175., -775.  -2050., -775.;
G54D12*G01X4620Y5300D02*G01X4820Y5300D01*G01X4820Y5350D01*G01X4495Y5350D01*G01X4
495Y5600D01*G01X4620Y5600D01*
ADD L19 :W25.
  2500., -1187.  2625., -1187.;
G54D14*G01X9170Y6012D02*G01X9295Y6012D01*
ADD L19 :W25.
  2487., -1325.  2487., -1200.;
G01X9157Y6150D02*G01X9157Y6025D01*
ADD L19 :W25.
  -3225., -1187.  -3100., -1187.;
G01X3445Y6012D02*G01X3570Y6012D01*
ADD L19 :W25.
  -3087., -1200.  -3087., -1325.;
G01X3583Y6025D02*G01X3583Y6150D01*
ADD L19 :W25.
  2625., 1687.  2500., 1687.;
```

•
•
•

```
G01X3170Y4625D02*G54D73*G55D03*
ADD R19 .'ZTARGET'  2894., 194.  2906., 206.;
G01X9570Y4625D02*G55D03*
ADD L5 :W45.
  -25., -1200.  -25., 125.  325., 125.;
G54D10*G01X6663Y6025D02*G01X6662Y4718D01*G01X6995Y4718D01*G01X6995Y4682D01*G01X6
628Y4682D01*G01X6627Y6025D01*G01X6663Y6025D01*G01X66
58Y6025D02*G01X6657Y4713D01*G01X6995Y4713D01*G01X6995Y4687D01*G01X6633Y4687D01*G
01X6632Y6025D01*G01X6658Y6025D01*G01X6653Y6025D02*G01
X6652Y4708D01*G01X6995Y4708D01*G01X6995Y4692D01*G01X6638Y4692D01*G01X6637Y6025D0
1*G01X6653Y6025D01*G01X6648Y6025D02*G01X6647Y4703D01*
G01X6995Y4703D01*G01X6995Y4697D01*G01X6643Y4697D01*G01X6642Y6025D01*G01X6648Y602
5D01*G01X0Y0D02*M00*M02*
```

The Gerber commands in the virtual plot file are commands supported on Gerber 4300 and 4400 series photo plotters. A minimum number of these Gerber commands are utilized by VPLOT to produce the virtual plot file. This is done to keep conversions to a minimum when working with a photo plotter other than a Gerber. The commands and symbols used are listed and described below.

Gerber Command	Meaning
G54Dxx	Select the aperture on the aperture wheel with a Gerber select code specified by xx. For example, G54D10 selects the aperture on the aperture wheel whose Gerber select code is 10.
G01XxxxxYyyyyD02	Move from the present x,y location to the x,y location specified by xxxx,yyyy AND keep the shutter closed during the move (i.e., move but do not plot while moving).
G01XxxxxYyyyyD01	Move from the present x,y location to the x,y location specified by xxxx,yyyy AND keep the shutter open during the move (i.e., trace a line from the present location to xxxx,yyyy).
G55D03	Flash the current aperture at the current location.
*	End of Gerber command.
M00	Program Stop.
M02	End of Program.

Creating Photo Plotter Mag Tape

Selecting Create Photoplotter Mag Tape from the Manager menu allows you to access the PPLOT program. PPLOT is a BASIC language program that extracts the Graphics Editor ADD commands used to create the drawing thus writing only the Gerber commands. If used, the HP 7970E must be present in the system configuration as peripheral #10 before selecting this function.

Selecting Create Photoplotter Mag Tape causes the system to prompt you for the file name where you wish the photo plotter data to be stored. Responding `TAPE:` causes the data to be written to the HP 7970E mag tape drive. Responding `PUNCH:` causes the data to be written to the HP 9884 paper tape punch. Any other response causes a BASIC data file to be created and the data stored there. For example, responding `MAG:F8` causes the photo plotter data to be stored in a BASIC data file named MAG on the mass storage device with msus = F8.

The system next prompts you for the file specifier identifying the virtual plot file that you wish to photo plot. The file specifier consists of an optional volume name or volume number (the default volume is the current system volume) and the complete EW file name of the virtual plot file (including any suffixes). If no file specifier is entered, the system again prompts you for the file name where you wish the photo plotter data to be stored. If PPLOT cannot find the volume or file specified, it returns an error message stating that the file/volume does not exist. It then again prompts you for the file specifier identifying the virtual plot file to be photo plotted.

Modifying the PPLOT Program

Because the PPLOT program is written in the BASIC programming language, you may modify it to provide other photo plotter data formats. The program has been written in a modular fashion (using subprograms to segment the tasks) and fully commented to help you understand what areas you need to modify. A thorough understanding of the BASIC programming language, and in particular, of subprograms, parameter passing, and I/O statements is necessary before attempting to modify the program. Familiarity with the HP 7970E Utility programs is also helpful if you intend to produce photo plotter data on the HP 7970E.

Designing Large Boards

The P. C. Layout Module is designed to handle P. C. boards of an average size of 80 square inches with a density of approximately 40 integrated circuits, although this may vary depending on the memory configuration and mass storage capabilities of your system. Much larger boards may be designed on the system if proper techniques are used.

One way to minimize the storage capacity needed for a board design is to make use of the hierarchial design capabilities of the Graphics Editor. Use library parts whereever possible as they always take up less space on your mass storage device and generally conserves computer memory. Not only does this save space, but it also speeds entry of the design. Regardless of the space saving techniques used, there may be boards which are either clumsy or impossible to edit because of their size and complexity. One method of using this system to create those boards involves designing the board in sections.

Designing Boards in Sections

A good practice, even for medium sized boards, is to design the board in sections. To do this, first enter the board outline, targets, notes, tooling holes, and other "overhead" components as a skeleton drawing. This skeleton can then be used along with each board section to assure accurate placement of each section on the board. The board can then be broken into sections, using either functional criteria or the smallest number of interconnects as the basis for the breakdown. Each section should be entered on system CRT in the appropriate location and edited until you are satisfied that it is reasonably complete. Next, using the WRAP command, the entire section is made into a single device. Be sure to note the origin you assign to the part when wrapping as you may need the exact coordinates later.

If you are designing a board which is likely to exceed the capacity of the system, save this version of the design using the SAVE command and a unique name for the drawing. Next, add interconnect traces to guide you in entering the next section of the design. Now, delete the device you just wrapped with the DELETE command leaving only the interconnect traces on your drawing (make sure that the device has been saved first). At this point you may enter the next section. When it is reasonably complete, it too can be wrapped and saved. Proceeding in this step-wise manner, you may enter each section. This limits the detail that the graphics editor has to deal with to the section currently being worked on. Once all sections of the design are complete the board may be "assembled" by adding instances of each device (design sections) that you created.

If the total board is too large to be "assembled" in this manner, you may photo plot and generate the N/C drill information for each section of the design separately or utilize the capability of the VDRILL program and the VPLOT program to combine multiple generate files to create N/C drill information and/or a photo plot.

Pen Plots and Verification Plots

Pen plots (without fill) are used by some P. C. fabrication shops to verify placement of components in the board layout. For simple boards, pen plots (with fill) can actually be used to create the artwork masters for manufacturing the board. The Graphics Editor provides the capability to produce pen plots of any drawing on an HP 9872B/C/S/T plotter or an HP 7580 plotter through the PLOT command in the edit subsystem. The PLOT command reproduces the drawing on the plotter exactly as it appears on the CRT. It does not recognize the conventions utilized by the VPLOT and PPLOT programs and thus, the plots produced are not necessarily identical to those produced on the photo plotter. Use of the :F parameter with the PLOT command allows certain figures to be filled when plotted. Specifying the pen width with the PLOT command allows the filling to be performed accurately so that the width of the pen is taken into consideration when plotting with fill.

Unattended Drill/Photo Plotter Data Generation

This section assumes that you have read and have thoroughly understood the Command File chapter of this manual, Chapter 6.

Since the generation of the virtual photo plotter data is a time consuming process, you might want to consider using the Command File Facility to produce your data. This allows you to enter all of the commands used to generate photo plotter data into a command file and then execute that file at a time when the system is not needed for graphic entry. However, since the Command File Facility assumes that all commands entered into the command file are correct, no provisions are made for interaction or error correction. Therefore, you should be sure of the validity of the commands you enter when creating your command file.

If you desire to execute either the virtual drill program or the virtual plot program from a command file, special commands must be executed instructing the system to load the program desired. Only the VDRILL and VPLOT programs can be run from a command file since they are contained in the EW environment.

To select the VDRILL program from a command file, you must enter a special line of code:

```
.
.
.
5 XEWPC:VDRILL.CODE
.
.
.
```

This is equivalent to selecting Create Virtual N/C Drill File from the Manager Tree. Since the system executes all commands unconditionally, you must make sure that all prompts are anticipated and correctly answered. For example, when the VDRILL program is selected, the system prompts you for the file specifier identifying the generate file to be operated on. It then prompts you for the name that you wish to assign to the virtual drill file to store the results in. For example:

```
1 XEWPC:VDRILL.CODE
2 TEST.GEN
3 PUNC
4 1
5 23
6
7 y
8 Y
9
10 XGEDIT.CODE
.
.
.
```

The above command file loads the VDRILL program and specifies the generate file, TEST-.GEN, as the file to be operated on. This is in response to the prompt issued by the system when the VDRILL program is selected. Next, the command file specifies the virtual drill file PUNC, as the file to contain the output of the VDRILL program. This is in response to the prompt by the system for the name of the virtual drill file to store the output in. Lines 4 and 5 in the command file supply the layer numbers of the drawing which are to be included in the virtual drill file. Line 6 in the command file terminates the prompting sequence for layers to be included in the virtual drill file. Line 7 specifies y as the major axis of the board. Line 8 verifies the correctness of the input data. These are supplied as a response to the prompts generated by the system.

Note that line 9 in the command file above is blank. This terminates the prompting sequence for names of generate files to operate on. Line 7 provides the carriage return that tells the system that no more generate files are to be input.

Once the VDRILL program performs its task, any other module may be selected (such as the Graphics Editor). VDRILL cannot be run or selected from inside any other program or module. Thus, if the Graphics Editor is selected in the command file, it must be exited completely (i.e., the BYE command must be executed from the MAIN) before attempting to execute the VDRILL program or the VPLOT program.

To select the VPLOT program from a command file, you must enter a special line of code:

```
.
.
.
XEWPC:VPLOT.CODE
.
.
.
```

This is equivalent to selecting the Create Virtual Plot File from the Manager Tree. Once VPLOT is selected, any commands available at that level are available through the command file. Like selecting VDRILL above, care must be taken to insure that all prompts are answered. All parameters that are specified with the commands interactively (such as :W xxxx, :F, etc...) should be entered with the command in the command file.

.
.
.

An Example

The following example illustrates one possible sequence of steps that could be used to create P. C. board artwork. The sequence of steps that you employ will vary since no two designers lay out a board in the exact same fashion.

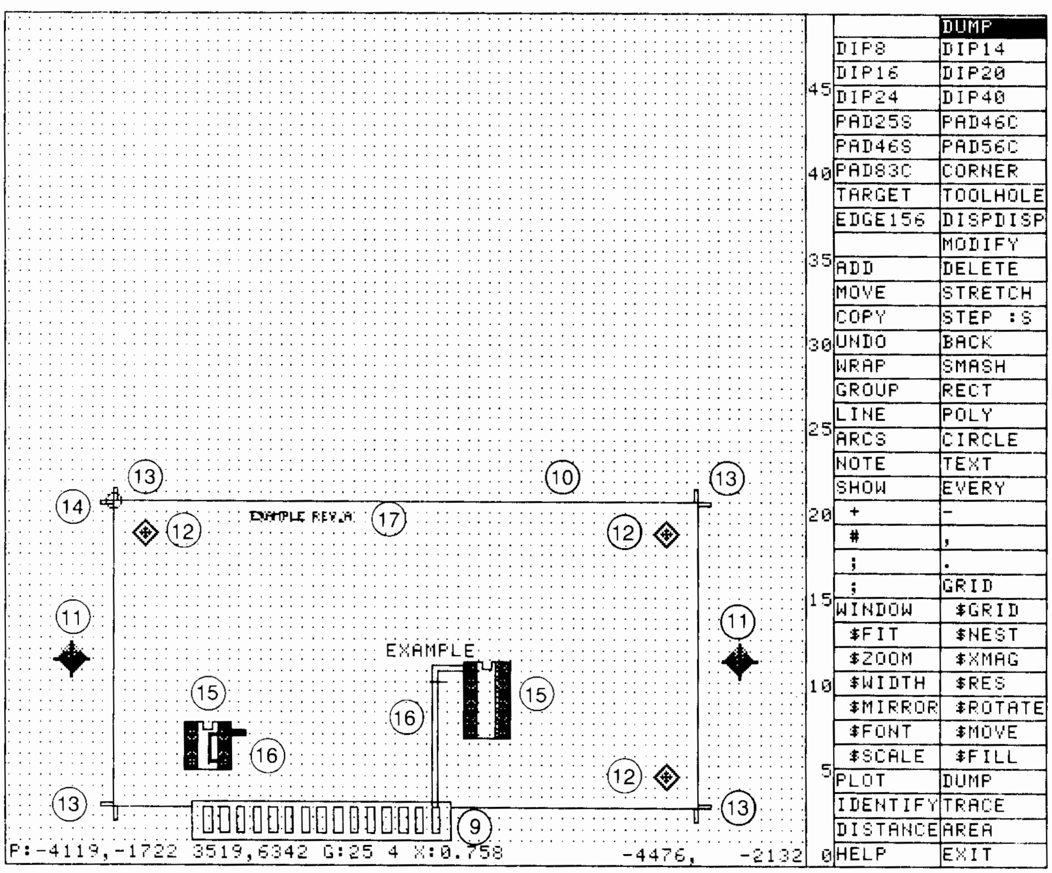
Before starting, you should create/add a new EW volume named EWLSC.

The numbers on the finished drawing at the end of this example correspond to the step adding that component to the drawing.

1. Load and run the EGS-45 software.
2. Select EGS-45 Graphics Editor from the P. C. Board Layout module in the Manager menu.

   Menu, macro, and process files which customize the edit subsystem for P. C. layout are loaded into the system at this time.
3. Enter the edit subsystem by entering:
   ```
   EDIT ;
   ```
4. Obtain a list of the current process file by entering:
   ```
   LIST PRO ;
   ```
5. Activate the display and modification of components on the display specific layers with the supplied DISPDISP macro:
   ```
   DISPDISP ;
   ```
6. Set the user grid such that a user grid point exists every 25 mils and such that every 4th grid point is displayed:
   ```
   GRID 25,4 ;
   ```
7. Turn on the display grid:
   ```
   GRID ;
   ```
8. Adjust the current window to zoom out by a factor of 15 and set the nesting depth displayed to 3:
   ```
   WINDOW $ZOOM -15  $NEST 3 ;
   ```
9. Add an edge connector:
   ```
   ADD  EDGE156 -2200, -1500 ;
   ```
10. Add the board blank:
    ```
    ADD LINE 2 300,-1200  2500,-1200  2500,1700  -3100,1700
    -3100,-1200  -2500,-1200 ;
    ```
11. Add the targets:
    ```
    ADD TARGET 2900,200 ;
    -3500,200 ;
    ```
12. Add the toolholes:
    ```
    ADD TOOLHOLE 2200,-900 ;
    2200,1400 ;
    -2800,1400 ;
    ```

13. Add the shear marks to each corner of the board. Since only one orientation of the shear mark is supplied, the supplied device is rotated to provide the needed orientations:
```
ADD CORNER 2500,-1200 ;
$ROTATE  90   2500,1700 ;
$ROTATE 180  -3100,1700 ;
$ROTATE -90  -3100,-1200 ;
```
14. Add the circle which acts as a reference location for the drill programs to the upper left corner of the P. C. board:
```
ADD CIRCLE 1 ,ORIGIN  -3100,1700  -3050,1750 ;
```
15. Add the IC packages to the board:
```
ADD DIP8  -2350,-475 ;
ADD DIP14  325,125 ;
```
16. Add the traces:
```
ADD LINE 5 $WIDTH 45  -25,-1200  -25,125  325,125 ;
$WIDTH 15  -2050,-475  -1850,-475  -1850,-525  -2175,-525
-2175,-775  -2050,-775 ;
```
17. Add a board title to the drawing:
```
ADD NOTE 19 $FONT 100 "EXAMPLE REV.A" -1800,1500 ;
```
18. Save the drawing:
```
SAVE EWLSC:EXAMPLE;
```
19. Obtain a printed copy of the drawing for a quick reference:
```
DUMP ;
```



20. Exit the edit subsytem:

```
EXIT;
```

21. Create a generate file from the drawing. The generate file should include all layers to be plotted, all drill layers, and all nesting levels:

```
GENERATE -E2 -E3 -E20 -E26 -E29 -E30 EWLSC:EXAMPLE EWLSC:;
```

This creates a generate file named EXAMPLE and stores the file on the volume EWLSC, and excludes layers that are never photo plotted.

22. Leave the Graphics Editor:

```
BYE
```

23. Select Create Virtual Plot File from the Manager menu.

24. Next, two virtual plot files are created, one for the metalization on the component side of the board and one for the solder resist on the component side.

Metalization on the component side includes the following layers: layer 5 (component metalization), layer 27 (pad master), and layer 19 (board reference and registration). The photo plot is offset 3000 mils in both the X and Y directions. To decrease the amount of time required for the actual plot, the components are sorted when the virtual plot file is created. By including the :I parameter, a small report is printed when the file is created. The :F parameter specifies to fill all circles, rectangles and lines (with width).

```
VPLOT :OX3000 :OY3000 :SY :F :I :L5 :L27 :L19 EWLSC:EXAMPLE,
EWLSC:EX,COMP
```

A virtual plot file named EX.COMP is created on the volume EWLSC. The printed report is shown below:

```
 Apertures used for file EWLSC:EXAMPLE.GEN

     Aperture                            Size      Uses
ZW10            Trace Circle             10         1
ZW12            Trace Circle             12         1
ZW15            Trace Circle             15         1
ZW25            Trace Circle             25         8
ZW80            Trace Circle             80        15
ZS55            Flash Square             55         4
ZOVAL           Flash Rectangle     75,  55        20
ZDIAM           Flash Square              0         3
ZTARGET         Flash Square              0         2

Wheel file used: EWPC:WHEEL1                  Error range:    plus/minus 0
                                              X offset used:           3000
                                              Y offset used:           3000
Input File: EWLSC:EXAMPLE.GEN                 Origin:                -3670,1825
Output File: EWLSC:EX.COMP                    Rectangles flashed:      26
                                              Rectangles filled:       0
Options set: Sort (Y)  Info  Fill             Total rectangles:        26
                                              Circles flashed:         3
                                              Circles filled:          0
Layers Plotted:                               Total circles:           3
5 19 27                                       Lines traced:            24
                                              Lines filled:            1
                                              Total lines:             25
                                              Polygons outlined:       0
                                              Arcs traced:             0
                                              Characters of text:      13
                                              Special names flashed:   25
                                              Null names thrown away:  10
                                              Draws:                    116.0
                                              Moves:                     69.0
                                              Flashes:                 29
                                              Aperture changes:        9

                                              Total components output: 55
```

The virtual plot file of the solder resist for the component side of the board should include the following layers: layer 28 (solder resist master), layer 21 (component side solder resist), and layer 19 (board reference and registration). The photo plot is offset 3000 mils in both the X and Y directions. Again, the components are sorted and filled. A short report is generated on the built-in printer.

```
VPLOT :OX3000 :OY3000 :S :F :I :L28 :L21 :L19 EWLSC:EXAMPLE,
EWLSC:EX.SLDR
```

A virtual plot file name EX.SLDR is created on the volume EWLSC. The printed report is shown below:

```
    Apertures used for file EWLSC:EXAMPLE.GEN

        Aperture                               Size      Uses
ZW12                 Trace Circle              12         1
ZW20                 Trace Circle              20         2
ZW25                 Trace Circle              25         8
ZW80                 Trace Circle              80        23
ZTARGET              Flash Square               0         2

Wheel file used: EWPC:WHEEL1


Input File: EWLSC:EXAMPLE.GEN
Output File: EWLSC:EX.SLDR

Options set: Sort (Y)  Info  Fill


Layers Plotted:
19 21 28

Error range:    plus/minus 0
X offset used:              3000
Y offset used:              3000
Origin:                  -3670,1825
Rectangles flashed:      2
Rectangles filled:       2
Total rectangles:        4
Circles flashed:         0
Circles filled:          3
Total circles:           3
Lines traced:            28
Lines filled:            0
Total lines:             28
Polygons outlined:       0
Arcs traced:             0
Characters of text:      13
Special names flashed:   2
Null names thrown away:  10
Draws:                     157.0
Moves:                      56.0
Flashes:                 2
Aperture changes:        7

Total components output: 36
```

25. If other photo plots are needed for other board sides, the above procedure would be repeated for each plot. After all needed layers have been processed, terminate the VPLOT program by entering:

```
BYE
```

26. Next, create the virtual drill files that will be needed to create the N/C data to produce the board. To do this, select Create Virtual Drill File from the Manager menu. The computer prompts and the responses to those prompts are shown below.

    Enter the name of the Generate file (CONT to leave).

    `EWLSC:EXAMPLE`

    Enter the name of the (output) virtual drill file.

    `EWLSC:EX.DRL`

    Enter a layer to take the drill data from (CONT when done).

    `1`

    Enter a layer to take the drill data from (CONT when done).

    (CONTINUE)

    Enter the major axis of the board (X or Y).

    `Y` (CONTINUE)

    Everything OK?

    `Y`

    Enter the name of the next generate file (CONT if none).

    (CONTINUE)

    The program now creates the virtual drill file.

    Enter the name of the Generate file (CONT to leave).

    (CONTINUE)

27. Create the final photo plotter data by selecting Create Photoplotter Mag Tape from the Manager menu. The prompts generated by the computer and the response to those prompts are shown below:

    Enter the name of the output file (File name/TAPE:/PUNCH:/null)

    `PLOT1:Q6`

    Enter the name of the virtual plot file (enter null to leave)

    `EWLSC:EX.COMP`

    Enter the name of the output file (File name/TAPE:/PUNCH:/null)

    `PLOT2:Q6`

    Enter the name of the virtual plot file (enter null to leave)

    `EWLSC:EX.SLDR`

    Enter the name of the output file (File name:TAPE:/PUNCH:/null)

    (CONTINUE)

    The photo plotter data now resides in two BASIC data files, PLOT1 and PLOT2, on the mass storage device whose msus is :Q6.

28. To create the final N/C drill data, select Create N/C Drill Paper Tape from the Manager menu. The computer prompts and the response to those prompts are shown below:

    Enter the name of the output file (File name/TAPE:/PUNCH:/null)

    `DRILL:Q6`

Enter the name of the virtual drill file (enter null to leave)

```
EWLSC:EX.DRL

Tool  1 -- Hole Size = 40          Number of holes =  22
Tool  2 -- Hole Size = 119         Number of holes =  3

Total Holes =  25
```

Enter the name of the output file (File name/TAPE:/PUNCH:/null)

CONTINUE

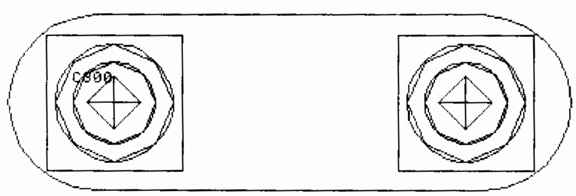
The N/C drill data (Excellon format) has been created. It is stored in a BASIC data file named DRILL on the mass storage device whose msus is :Q6.

A report is output to the system printer that is a summary of the drill information. This report is used to set up the N/C drill machine.

Supplied Library Parts

The following section contains representations of the supplied library parts. A listing of the archive file produced from each supplied device is also provided. This shows the commands of the Graphics Editor and its subsystems used to create each library part.

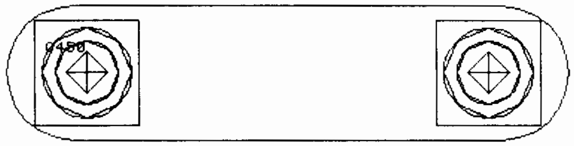
C300




```
$FILES  C300, PAD46C, LP46C $;
EDIT LP46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -50,-50  50,50;
ADD C27 :R6.00  0,0  35,0;
ADD C28 :R6.00  0,0  50,0;
ADD C29 :R45.00  0,0  35,0;
ADD C3 :R6.00  0,0  35,0;
ADD C30 :R45.00  0,0  50,0;
ADD C20 :R6.00  0,0  50,0;
ADD L1 :W0  -23,0  23,0;
ADD L1 :W0  0,-23  0,23;
ADD C1 .'D0046' :R90.00  0,0  23,0;
GRID 5,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

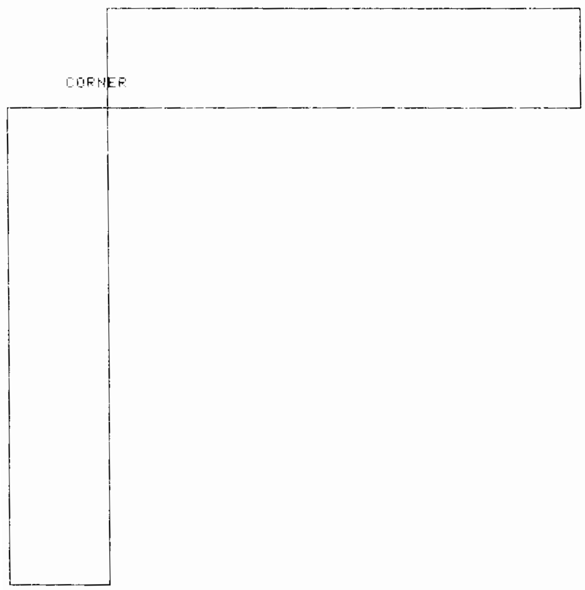
```
EDIT PAD46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -58,-58  58,58;
ADD LP46C  0,0;
ADD R4  -58,-58  58,58;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT C300;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -90,-75  390,75;
ADD A26 :R15.00  315,-75  315,75  390,0;
ADD A26 :R15.00  -15,75  -15,-75  -90,0;
ADD L26 :W0  -15,-75  315,-75;
ADD L26 :W0  315,75  -15,75;
ADD PAD46C  300,0;
ADD PAD46C  0,0;
GRID 5,5  0,0;
LOCK 15.00;
SAVE;
EXIT;
```

C450



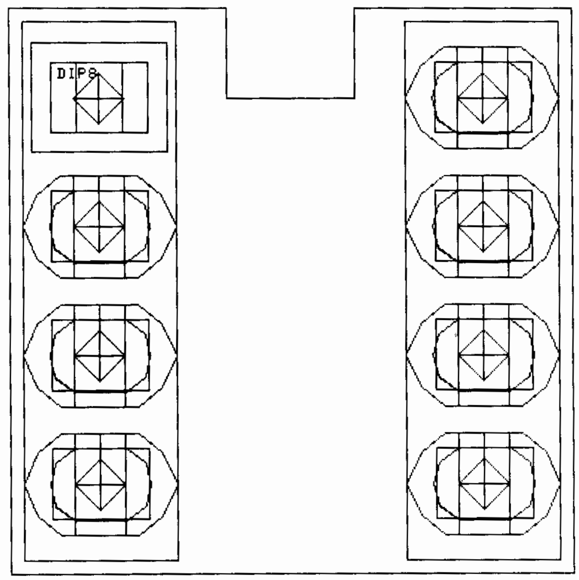

```
$FILES  C450, PAD46C, LP46C $;
EDIT LP46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -50,-50  50,50;
ADD C27 :R6.00  0,0  35,0;
ADD C28 :R6.00  0,0  50,0;
ADD C29 :R45.00  0,0  35,0;
ADD C3 :R6.00  0,0  35,0;
ADD C30 :R45.00  0,0  50,0;
ADD C20 :R6.00  0,0  50,0;
ADD L1 :W0  -23,0  23,0;
ADD L1 :W0  0,-23  0,23;
ADD C1 .'D0046' :R90.00  0,0  23,0;
GRID 5,1  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT PAD46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -58,-58  58,58;
ADD LP46C  0,0;
ADD R4  -58,-58  58,58;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT C450;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -90,-75  540,75;
ADD L26 :W0  -15,-75  465,-75;
ADD L26 :W0  -15,75  465,75;
ADD A26 :R15.00  465,75  465,-75  540,0;
ADD A26 :R15.00  -15,75  -15,-75  -90,0;
ADD PAD46C  450,0;
ADD PAD46C  0,0;
GRID 5,5  0,0;
LOCK 15.00;
SAVE;
EXIT;
```

CORNER



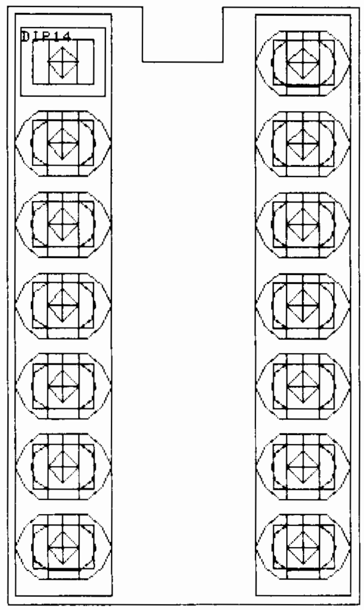

```
$FILES  CORNER $;
EDIT CORNER;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -26,-125  125,26;
ADD L19 :W25  0,13  125,13;
ADD L19 :W25  -13,-125  -13,0;
GRID 1,25  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

DIP8



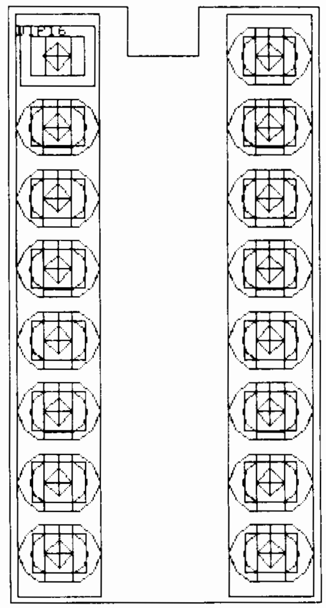
```
$FILES  DIP8, LP400, LP40R $;
EDIT LP400;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -60,-40  60,40;
ADD C1 .'D0040' :R90.00  0,0  20,0;
ADD P29  -20,28  21,28  36,16  40,0  36,-16  21,-28  -20,-28  -35,-17
  -39,0  -35,16  -20,28;
ADD P30  -30,40  30,40  49,24  60,0  49,-24  30,-40  -30,-40  -49,-24
  -60,0  -49,24  -30,40;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD L3 :W28  0,14  16,14  24,8  26,0  24,-8  16,-14  -15,-14  -23,-8
  -25,0  -23,8  -15,14  0,14;
ADD R27 .'ZOVAL'  -37,-27  38,28;
ADD L28 :W80  -20,0  20,0;
ADD L20 :W40  0,20  23,20  33,12  38,0  33,-12  23,-20  -23,-20
  -33,-12  -38,0  -33,12  -23,20  0,20;
GRID 1,1  0,0;
LOCK 15.00;
SAVE;
EXIT;

EDIT LP40R;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -52,-42  53,43;
ADD R28  -52,-42  53,43;
ADD R30  -52,-42  53,43;
ADD R20  -52,-42  53,43;
ADD R29  -37,-27  38,28;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD R3  -37,-27  38,28;
ADD R27  -37,-27  18,28;
ADD R27  -17,-27  38,28;
ADD C1 .'D0040' :R90.00  0,0  20,0;
GRID 5,1  -37,28;
LOCK 15.00;
SAVE;
EXIT;

EDIT DIP8;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -70,-370  370,70;
ADD L26 :W0  -70,70  -70,-370  370,-370  370,70  200,70  200,0
  100,0  100,70  -70,70;
ADD LP400  300,0;
ADD LP400  0,-100;
ADD LP400  300,-100;
ADD LP400  0,-200;
ADD LP400  300,-200;
ADD LP400  0,-300;
ADD LP400  300,-300;
ADD R4  240,-360  360,60;
ADD R4  -60,-360  60,60;
ADD LP40R  0,0;
GRID 20,2  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

DIP14



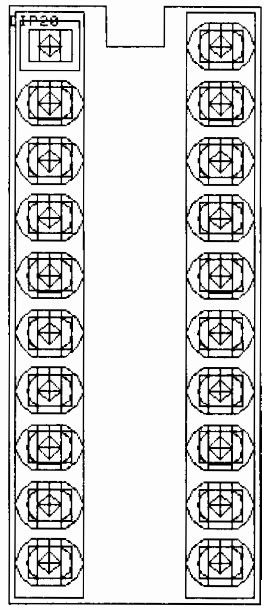
```
$FILES  DIP14, LP400, LP40R $;
EDIT LP400;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -60,-40  60,40;
ADD C1 .'D0040' :R90.00  0,0  20,0;
ADD P29  -20,28  21,28  36,16  40,0  36,-16  21,-28  -20,-28  -35,-17
  -39,0  -35,16  -20,28;
ADD P30  -30,40  30,40  49,24  60,0  49,-24  30,-40  -30,-40  -49,-24
  -60,0  -49,24  -30,40;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD L3 :W28  0,14  16,14  24,8  26,0  24,-8  16,-14  -15,-14  -23,-8
  -25,0  -23,8  -15,14  0,14;
ADD R27 .'ZOVAL'  -37,-27  38,28;
ADD L28 :W80  -20,0  20,0;
ADD L20 :W40  0,20  23,20  33,12  38,0  33,-12  23,-20  -23,-20
  -33,-12  -38,0  -33,12  -23,20  0,20;
GRID 1,1  0,0;
LOCK 15.00;
SAVE;
EXIT;

EDIT LP40R;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -52,-42  53,43;
ADD R28  -52,-42  53,43;
ADD R30  -52,-42  53,43;
ADD R20  -52,-42  53,43;
ADD R29  -37,-27  38,28;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD R3  -37,-27  38,28;
ADD R27  -37,-27  18,28;
ADD R27  -17,-27  38,28;
ADD C1 .'D0040' :R90.00  0,0  20,0;
GRID 5,1  -37,28;
LOCK 15.00;
SAVE;
EXIT;

EDIT DIP14;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -70,-670  370,70;
ADD L26 :W0  -70,70  -70,-670  370,-670  370,70  200,70  200,0
  100,0  100,70  -70,70;
ADD LP400  300,0;
ADD LP400  0,-100;
ADD LP400  300,-100;
ADD LP400  0,-200;
ADD LP400  300,-200;
ADD LP400  0,-300;
ADD LP400  300,-300;
ADD LP400  0,-400;
ADD LP400  300,-400;
ADD LP400  0,-500;
ADD LP400  300,-500;
ADD LP400  0,-600;
ADD LP400  300,-600;
ADD R4  -60,-660  60,60;
ADD R4  240,-660  360,60;
ADD LP40R  0,0;
GRID 20,2  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

DIP16



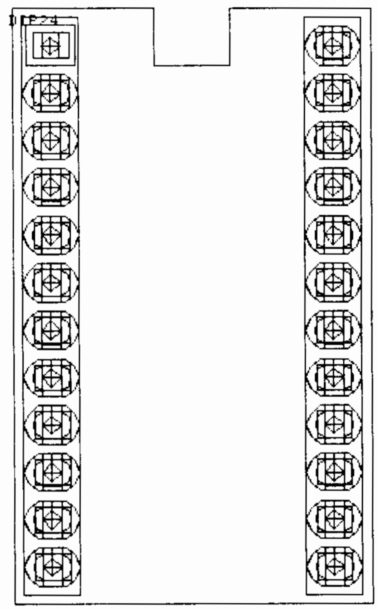
```
$FILES  DIP16, LP400, LP40R $;
EDIT LP400;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -60,-40  60,40;
ADD C1 .'D0040' :R90.00  0,0  20,0;
ADD P29  -20,28  21,28  36,16  40,0  36,-16  21,-28  -20,-28  -35,-17
  -39,0  -35,16  -20,28;
ADD P30  -30,40  30,40  49,24  60,0  49,-24  30,-40  -30,-40  -49,-24
  -60,0  -49,24  -30,40;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD L3 :W28  0,14  16,14  24,8  26,0  24,-8  16,-14  -15,-14  -23,-8
  -25,0  -23,8  -15,14  0,14;
ADD R27 .'ZOVAL'  -37,-27  38,28;
ADD L28 :W80  -20,0  20,0;
ADD L20 :W40  0,20  23,20  33,12  38,0  33,-12  23,-20  -23,-20
  -33,-12  -38,0  -33,12  -23,20  0,20;
GRID 1,1  0,0;
LOCK 15.00;
SAVE;
EXIT;

EDIT LP40R;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -52,-42  53,43;
ADD R28  -52,-42  53,43;
ADD R30  -52,-42  53,43;
ADD R20  -52,-42  53,43;
ADD R29  -37,-27  38,28;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD R3  -37,-27  38,28;
ADD R27  -37,-27  18,28;
ADD R27  -17,-27  38,28;
ADD C1 .'D0040' :R90.00  0,0  20,0;
GRID 5,1  -37,28;
LOCK 15.00;
SAVE;
EXIT;

EDIT DIP16;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -70,-770  370,70;
ADD L26 :W0  -70,70  -70,-770  370,-770  370,70  200,70  200,0
  100,0  100,70  -70,70;
ADD LP400  300,0;
ADD LP400  0,-100;
ADD LP400  300,-100;
ADD LP400  0,-200;
ADD LP400  300,-200;
ADD LP400  0,-300;
ADD LP400  300,-300;
ADD LP400  0,-400;
ADD LP400  300,-400;
ADD LP400  0,-500;
ADD LP400  300,-500;
ADD LP400  0,-600;
ADD LP400  300,-600;
ADD LP400  0,-700;
ADD LP400  300,-700;
ADD R4  -60,-760  60,60;
ADD R4  240,-760  360,60;
ADD LP40R  0,0;
GRID 20,2  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

DIP20



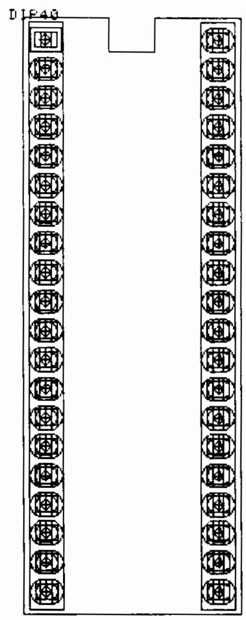
```
$FILES  DIP20, LP400, LP40R $;
EDIT LP400;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -60,-40  60,40;
ADD C1 .'D0040' :R90.00  0,0  20,0;
ADD P29  -20,28  21,28  36,16  40,0  36,-16  21,-28  -20,-28  -35,-17
  -39,0  -35,16  -20,28;
ADD P30  -30,40  30,40  49,24  60,0  49,-24  30,-40  -30,-40  -49,-24
  -60,0  -49,24  -30,40;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD L3 :W28  0,14  16,14  24,8  26,0  24,-8  16,-14  -15,-14  -23,-8
  -25,0  -23,8  -15,14  0,14;
ADD R27 .'ZOVAL'  -37,-27  38,28;
ADD L28 :W80  -20,0  20,0;
ADD L20 :W40  0,20  23,20  33,12  38,0  33,-12  23,-20  -23,-20
  -33,-12  -38,0  -33,12  -23,20  0,20;
GRID 1,1  0,0;
LOCK 15.00;
SAVE;
EXIT;

EDIT LP40R;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -52,-42  53,43;
ADD R28  -52,-42  53,43;
ADD R30  -52,-42  53,43;
ADD R20  -52,-42  53,43;
ADD R29  -37,-27  38,28;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD R3  -37,-27  38,28;
ADD R27  -37,-27  18,28;
ADD R27  -17,-27  38,28;
ADD C1 .'D0040' :R90.00  0,0  20,0;
GRID 5,1  -37,28;
LOCK 15.00;
SAVE;
EXIT;

EDIT DIP20;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -70,-970  370,70;
ADD L26 :W0  -70,70  -70,-970  370,-970  370,70  200,70  200,0
  100,0  100,70  -70,70;
ADD LP400  0,-100;
ADD LP400  0,-200;
ADD LP400  0,-300;
ADD LP400  0,-400;
ADD LP400  0,-500;
ADD LP400  0,-600;
ADD LP400  0,-700;
ADD LP400  0,-800;
ADD LP400  0,-900;
ADD LP40R  0,0;
ADD R4  -60,-960  60,60;
ADD LP400  300,0;
ADD LP400  300,-100;
ADD LP400  300,-200;
ADD LP400  300,-300;
ADD LP400  300,-400;
ADD LP400  300,-500;
ADD LP400  300,-600;
ADD LP400  300,-700;
ADD LP400  300,-800;
ADD LP400  300,-900;
ADD R4  240,-960  360,60;
GRID 20,2  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

DIP24



```
$FILES  DIP24, LP400, LP40R $;
EDIT LP400;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -60,-40  60,40;
ADD C1 .'D0040' :R90.00  0,0  20,0;
ADD P29  -20,28  21,28  36,16  40,0  36,-16  21,-28  -20,-28  -35,-17
  -39,0  -35,16  -20,28;
ADD P30  -30,40  30,40  49,24  60,0  49,-24  30,-40  -30,-40  -49,-24
  -60,0  -49,24  -30,40;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD L3 :W28  0,14  16,14  24,8  26,0  24,-8  16,-14  -15,-14  -23,-8
  -25,0  -23,8  -15,14  0,14;
ADD R27 .'ZOVAL'  -37,-27  38,28;
ADD L28 :W80  -20,0  20,0;
ADD L20 :W40  0,20  23,20  33,12  38,0  33,-12  23,-20  -23,-20
  -33,-12  -38,0  -33,12  -23,20  0,20;
GRID 1,1  0,0;
LOCK 15.00;
SAVE;
EXIT;

EDIT LP40R;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -52,-42  53,43;
ADD R28  -52,-42  53,43;
ADD R30  -52,-42  53,43;
ADD R20  -52,-42  53,43;
ADD R29  -37,-27  38,28;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD R3  -37,-27  38,28;
ADD R27  -37,-27  18,28;
ADD R27  -17,-27  38,28;
ADD C1 .'D0040' :R90.00  0,0  20,0;
GRID 5,1  -37,28;
LOCK 15.00;
SAVE;
EXIT;

EDIT DIP24;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -80,-1180  680,80;
ADD L26 :W0  -80,80  -80,-1180  680,-1180  680,80  380,80  380,-40
  220,-40  220,80  -80,80;
ADD LP400  600,0;
ADD LP400  0,-100;
ADD LP400  600,-100;
ADD LP400  0,-200;
ADD LP400  600,-200;
ADD LP400  0,-300;
ADD LP400  600,-300;
ADD LP400  0,-400;
ADD LP400  600,-400;
ADD LP400  0,-500;
ADD LP400  600,-500;
ADD LP400  0,-600;
ADD LP400  600,-600;
ADD LP400  0,-700;
ADD LP400  600,-700;
ADD LP400  0,-800;
ADD LP400  600,-800;
ADD LP400  0,-900;
ADD LP400  600,-900;
ADD LP400  0,-1000;
ADD LP400  600,-1000;
ADD LP400  0,-1100;
ADD LP400  600,-1100;
ADD R4  -60,-1160  60,60;
ADD R4  540,-1160  660,60;
ADD LP40R  0,0;
GRID 20,2  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

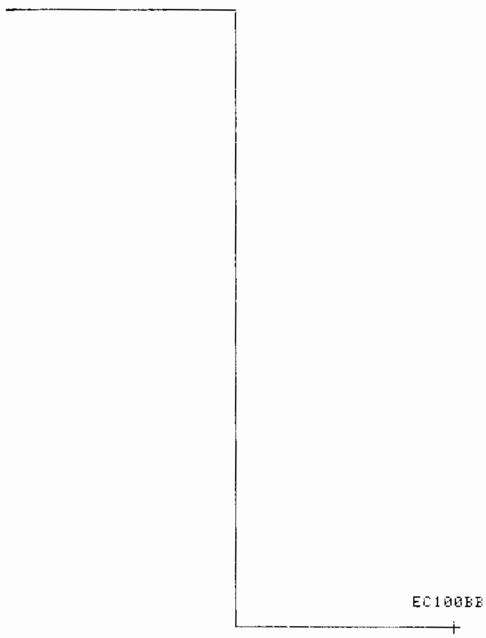
DIP40




```
$FILES  DIP40, LP400, LP40R $;
EDIT LP400;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -60,-40  60,40;
ADD C1 .'D0040' :R90.00  0,0  20,0;
ADD P29  -20,28  21,28  36,16  40,0  36,-16  21,-28  -20,-28  -35,-17
  -39,0  -35,16  -20,28;
ADD P30  -30,40  30,40  49,24  60,0  49,-24  30,-40  -30,-40  -49,-24
  -60,0  -49,24  -30,40;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD L3 :W28  0,14  16,14  24,8  26,0  24,-8  16,-14  -15,-14  -23,-8
  -25,0  -23,8  -15,14  0,14;
ADD R27 .'ZOVAL'  -37,-27  38,28;
ADD L28 :W80  -20,0  20,0;
ADD L20 :W40  0,20  23,20  33,12  38,0  33,-12  23,-20  -23,-20
  -33,-12  -38,0  -33,12  -23,20  0,20;
GRID 1,1  0,0;
LOCK 15.00;
SAVE;
EXIT;
```

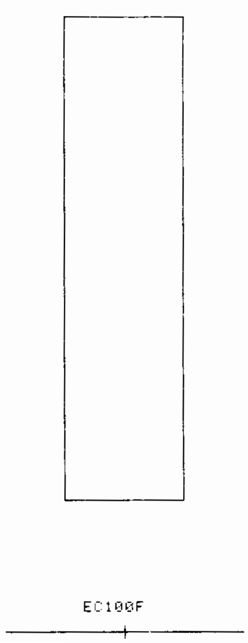
```
EDIT LP40R;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -52,-42  53,43;
ADD R28  -52,-42  53,43;
ADD R30  -52,-42  53,43;
ADD R20  -52,-42  53,43;
ADD R29  -37,-27  38,28;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD R3  -37,-27  38,28;
ADD R27  -37,-27  18,28;
ADD R27  -17,-27  38,28;
ADD C1 .'D0040' :R90.00  0,0  20,0;
GRID 5,1  -37,28;
LOCK 15.00;
SAVE;
EXIT;

EDIT DIP40;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -80,-1980  680,80;
ADD L26 :W0  -80,80  -80,-1980  680,-1980  680,80  380,80  380,-40
  220,-40  220,80  -80,80;
ADD LP400  600,0;
ADD LP400  0,-100;
ADD LP400  600,-100;
ADD LP400  0,-200;
ADD LP400  600,-200;
ADD LP400  0,-300;
ADD LP400  600,-300;
ADD LP400  0,-400;
ADD LP400  600,-400;
ADD LP400  0,-500;
ADD LP400  600,-500;
ADD LP400  0,-600;
ADD LP400  600,-600;
ADD LP400  0,-700;
ADD LP400  600,-700;
ADD LP400  0,-800;
ADD LP400  600,-800;
ADD LP400  0,-900;
ADD LP400  600,-900;
ADD LP400  0,-1000;
ADD LP400  600,-1000;
ADD LP400  0,-1100;
ADD LP400  600,-1100;
ADD LP400  0,-1200;
ADD LP400  600,-1200;
ADD LP400  0,-1300;
ADD LP400  600,-1300;
ADD LP400  0,-1400;
ADD LP400  600,-1400;
ADD LP400  0,-1500;
ADD LP400  600,-1500;
ADD LP400  0,-1600;
ADD LP400  600,-1600;
ADD LP400  0,-1700;
ADD LP400  600,-1700;
ADD LP400  0,-1800;
ADD LP400  600,-1800;
ADD LP400  0,-1900;
ADD LP400  600,-1900;
ADD R4  -60,-1960  60,60;
ADD R4  540,-1960  660,60;
ADD LP40R  0,0;
GRID 20,2  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

EC100BB



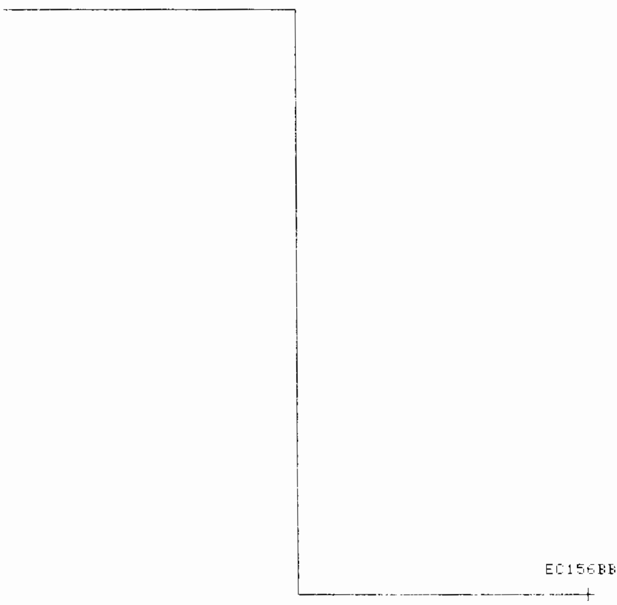

```
$FILES  EC100BB $;
EDIT EC100BB;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -200,0  0,250;
ADD L2 :W0  0,0  -90,0  -90,250  -200,250;
GRID 10,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

EC100F



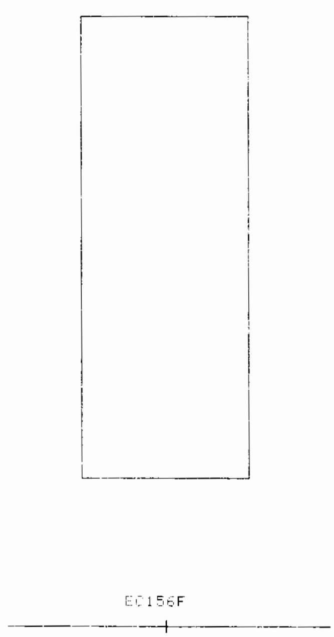

```
$FILES  EC100F $;
EDIT EC100F;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -50,0  50,255;
ADD P29  -25,55  25,55  25,255  -25,255  -25,55;
ADD L27 :W50  0,55  0,255;
ADD L2 :W0  -50,0  50,0;
ADD L3 :W50  0,55  0,255;
GRID 1,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

EC156BB



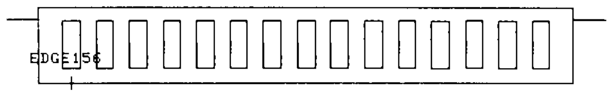

```
$FILES  EC156BB $;
EDIT EC156BB;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -300,0  0,300;
ADD L2 :W0  0,0  -150,0  -150,300  -300,300;
GRID 10,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

EC156F



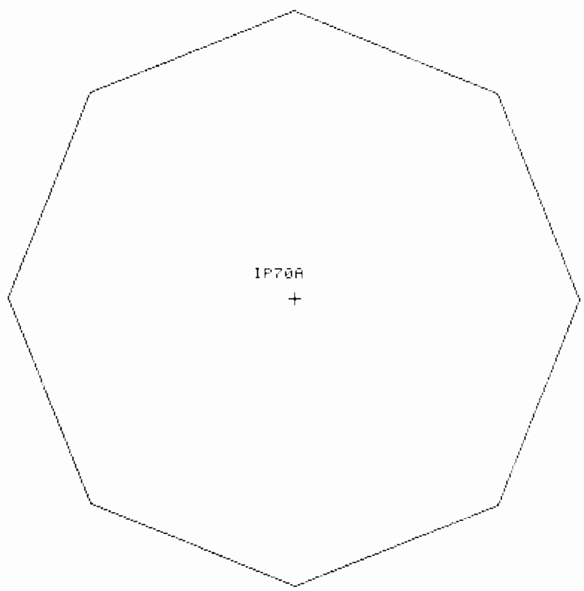

```
$FILES  EC156F $;
EDIT EC156F;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -78,0  78,290;
ADD P29  -40,70  40,70  40,290  -40,290  -40,70;
ADD L27 :W80  0,70  0,290;
ADD L2 :W0  -78,0  78,0;
ADD L3 :W80  0,70  0,290;
GRID 1,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

EDGE156



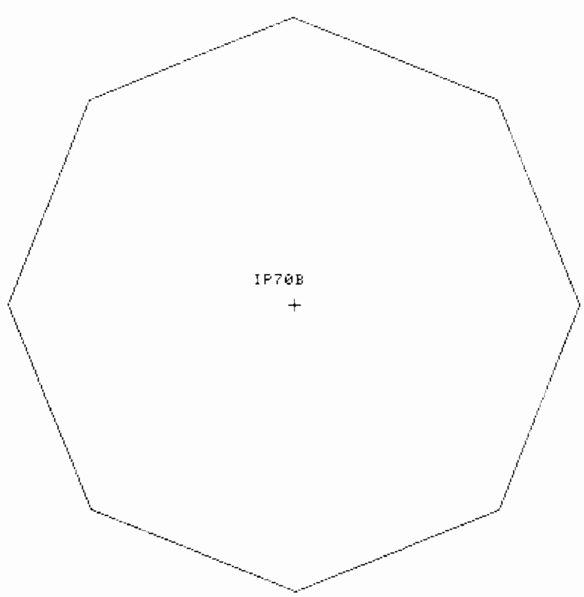

```
$FILES  EDGE156, EC156F, EC156BB $;
EDIT EC156F;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -78,0  78,290;
ADD P29  -40,70  40,70  40,290  -40,290  -40,70;
ADD L27 :W80  0,70  0,290;
ADD L2 :W0  -78,0  78,0;
ADD L3 :W80  0,70  0,290;
GRID 1,1  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT EC156BB;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -300,0  0,300;
ADD L2 :W0  0,0  -150,0  -150,300  -300,300;
GRID 10,1  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT EDGE156;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -300,0  2484,350;
ADD EC156F  0,0;
ADD EC156BB  0,0;
ADD EC156BB :MY  2184,0;
ADD EC156F  156,0;
ADD EC156F  312,0;
ADD EC156F  468,0;
ADD EC156F  624,0;
ADD EC156F  780,0;
ADD EC156F  936,0;
ADD EC156F  1092,0;
ADD EC156F  1248,0;
ADD EC156F  1404,0;
ADD EC156F  1560,0;
ADD EC156F  1716,0;
ADD EC156F  1872,0;
ADD EC156F  2028,0;
ADD EC156F  2184,0;
ADD R4  -150,0  2334,350;
GRID 156,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

IP70A



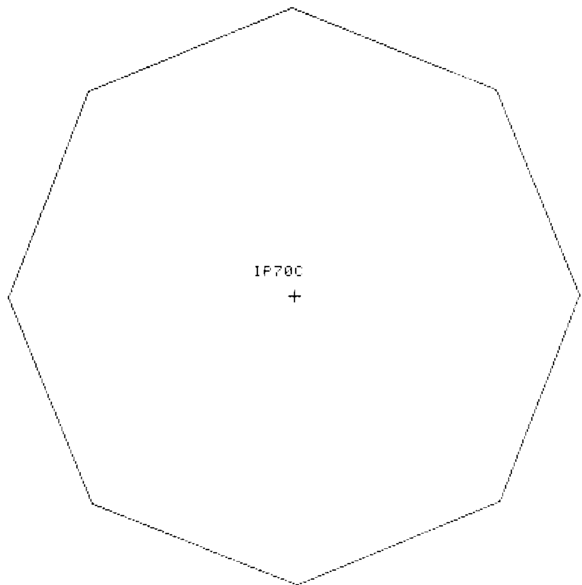

```
$FILES  IP70A $;
EDIT IP70A;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -35,-35  35,35;
ADD C11 :R45.00  0,0  35,0;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

IP70B



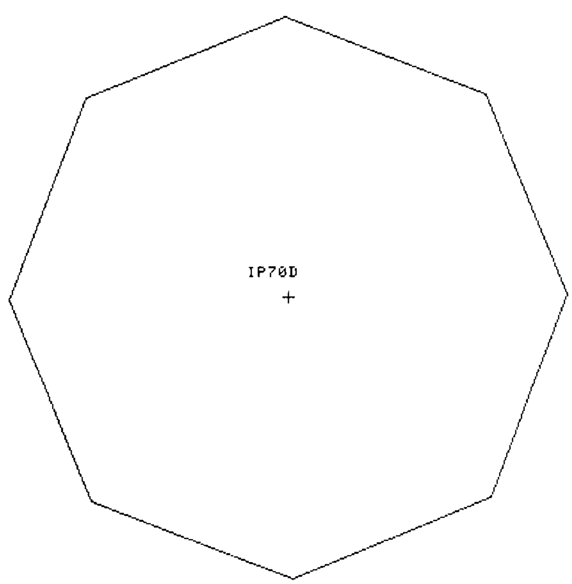

```
$FILES  IP70B $;
EDIT IP70B;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -35,-35  35,35;
ADD C12 :R45.00  0,0  35,0;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

IP70C



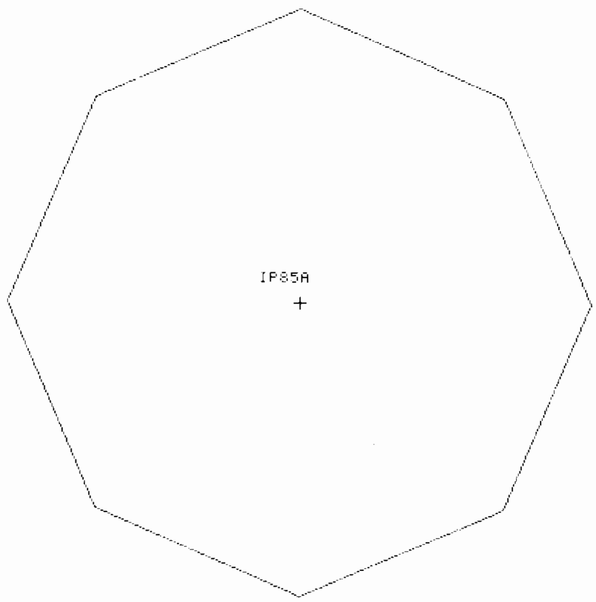

```
$FILES  IP70C $;
EDIT IP70C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -35,-35  35,35;
ADD C13 :R45.00  0,0  35,0;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

IP70D



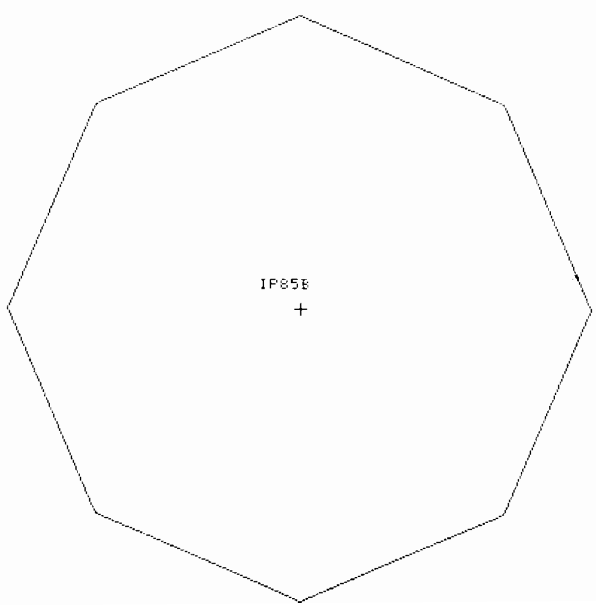

```
$FILES  IP70D $;
EDIT IP70D;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -35,-35  35,35;
ADD C14 :R45.00  0,0  35,0;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

IP85A



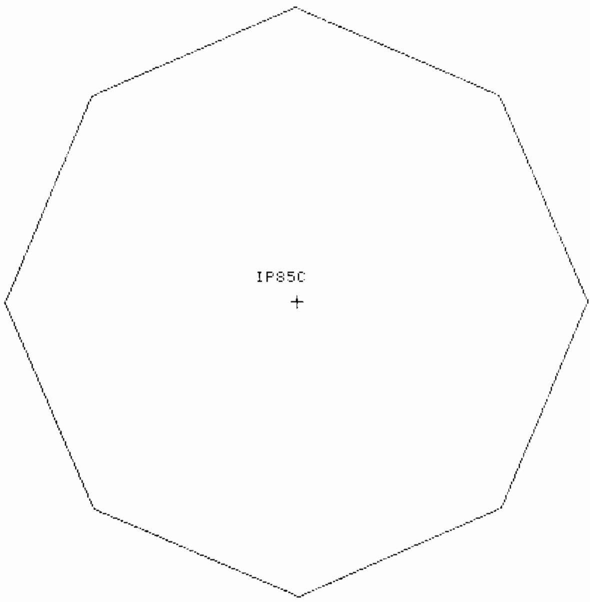

```
$FILES  IP85A $;
EDIT IP85A;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -43,-43  43,43;
ADD C11 :R45.00  0,0  43,0;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

IP85B



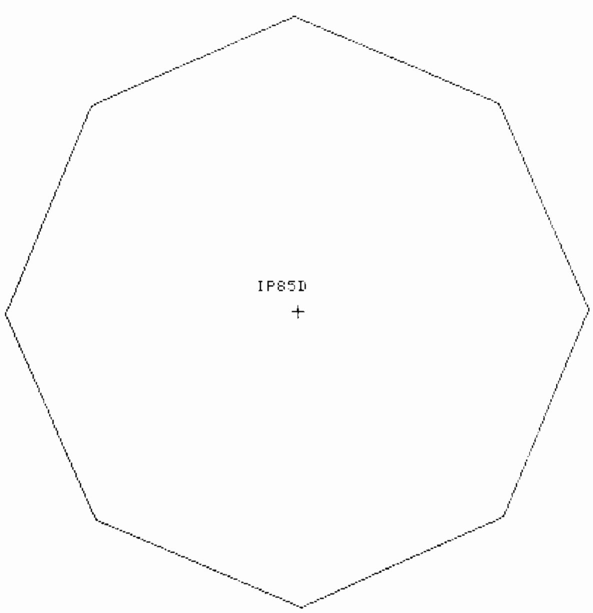

```
$FILES  IP85B $;
EDIT IP85B;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -43,-43  43,43;
ADD C12 :R45.00  0,0  43,0;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

IP85C



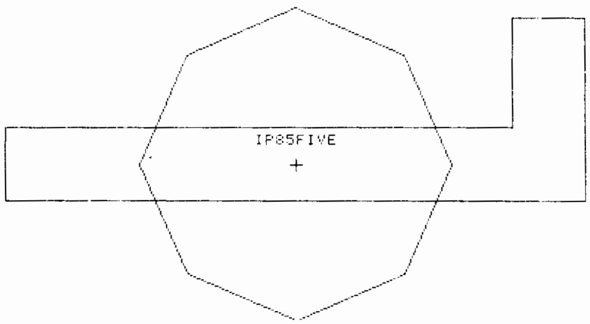

```
$FILES  IP85C $;
EDIT IP85C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -43,-43  43,43;
ADD C13 :R45.00  0,0  43,0;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

IP85D



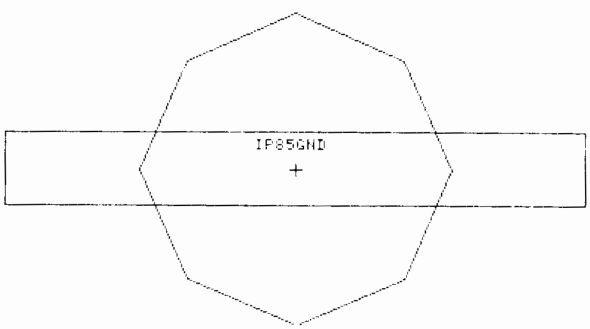

```
$FILES  IP85D $;
EDIT IP85D;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -43,-43  43,43;
ADD C14 :R45.00  0,0  43,0;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

IP85FIVE



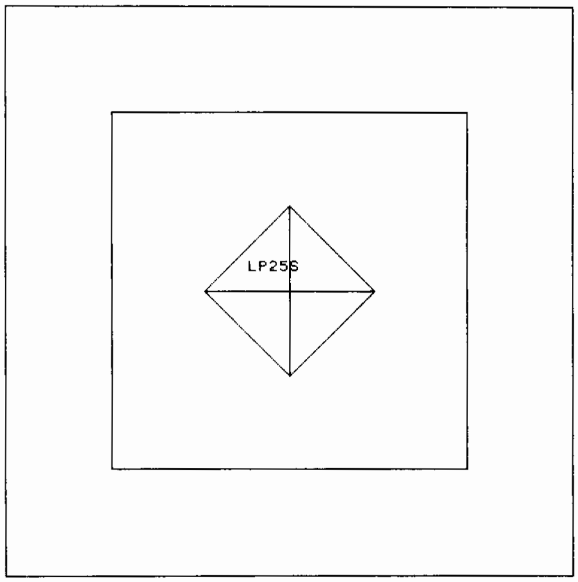

```
$FILES  IP85FIVE $;
EDIT IP85FIVE;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -80,-43  80,43;
ADD C7 :R45.00  0,0  43,0;
ADD L7 :W20  -80,0  70,0  70,40;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

IP85GND



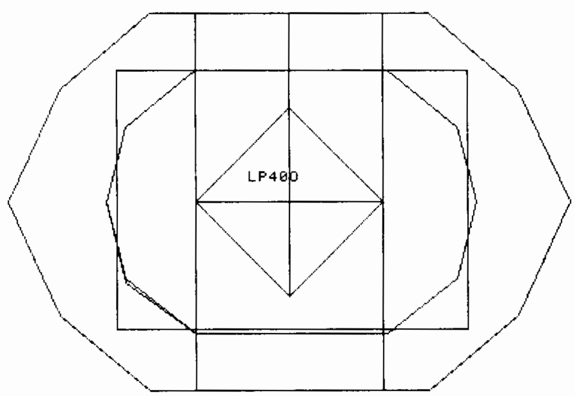

```
$FILES  IP85GND $;
EDIT IP85GND;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -80,-43  80,43;
ADD C8 :R45.00  0,0  43,0;
ADD L8 :W20  -80,0  80,0;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

LP25S



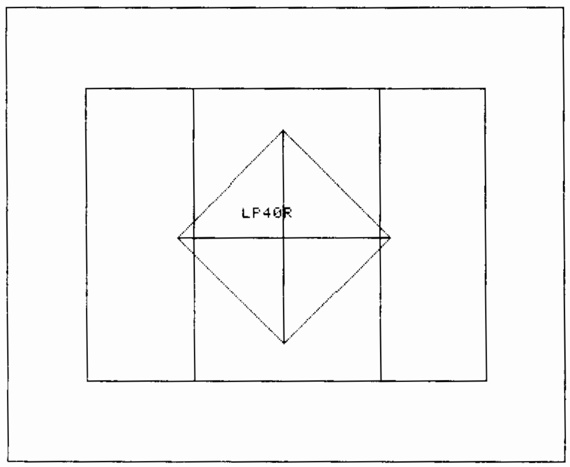

```
$FILES  LP25S $;
EDIT LP25S;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -40,-40  40,40;
ADD R28  -40,-40  40,40;
ADD R30  -40,-40  40,40;
ADD R20  -40,-40  40,40;
ADD R29  -25,-25  25,25;
ADD R27  -25,-25  25,25;
ADD L1 :W0  -12,0  12,0;
ADD L1 :W0  0,-12  0,12;
ADD C1 .'D0025' :R90.00  0,0  12,0;
ADD R3  -25,-25  25,25;
GRID 5,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

LP4Ø0



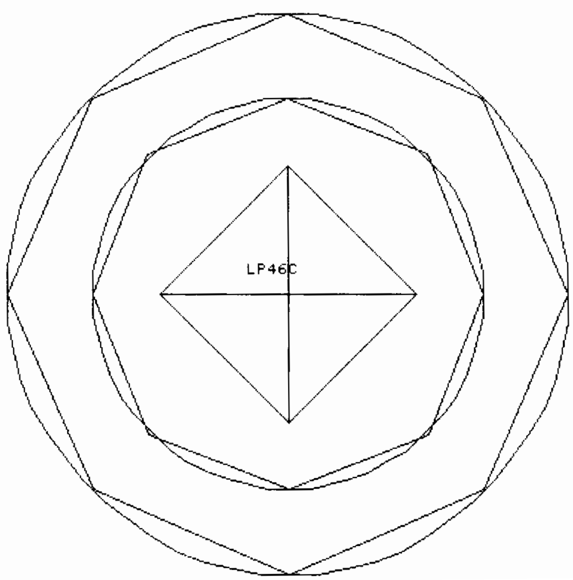

```
$FILES  LP400 $;
EDIT LP400;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -60,-40  60,40;
ADD C1 .'D0040' :R90.00  0,0  20,0;
ADD P29  -20,28  21,28  36,16  40,0  36,-16  21,-28  -20,-28  -35,-17
  -39,0  -35,16  -20,28;
ADD P30  -30,40  30,40  49,24  60,0  49,-24  30,-40  -30,-40  -49,-24
  -60,0  -49,24  -30,40;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD L3 :W28  0,14  16,14  24,8  26,0  24,-8  16,-14  -15,-14  -23,-8
  -25,0  -23,8  -15,14  0,14;
ADD R27 .'ZOVAL'  -37,-27  38,28;
ADD L28 :W80  -20,0  20,0;
ADD L20 :W40  0,20  23,20  33,12  38,0  33,-12  23,-20  -23,-20
  -33,-12  -38,0  -33,12  -23,20  0,20;
GRID 1,1  0,0;
LOCK 15.00;
SAVE;
EXIT;
```

LP40R



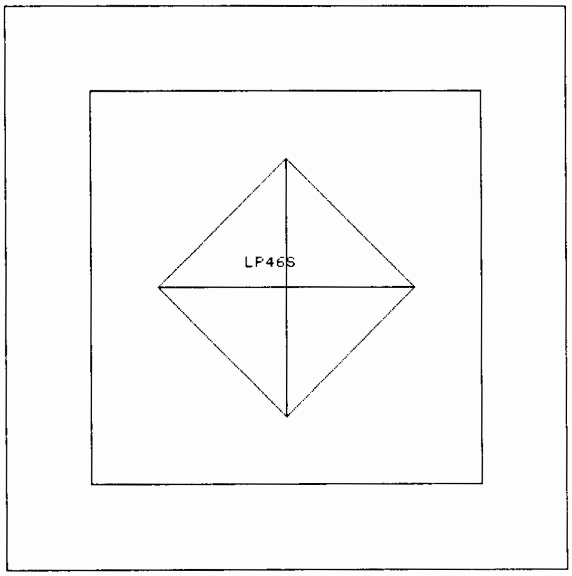

```
$FILES  LP40R $;
EDIT LP40R;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -52,-42  53,43;
ADD R28  -52,-42  53,43;
ADD R30  -52,-42  53,43;
ADD R20  -52,-42  53,43;
ADD R29  -37,-27  38,28;
ADD L1 :W0  -20,0  20,0;
ADD L1 :W0  0,20  0,-20;
ADD R3  -37,-27  38,28;
ADD R27  -37,-27  18,28;
ADD R27  -17,-27  38,28;
ADD C1 .'D0040' :R90.00  0,0  20,0;
GRID 5,1  -37,28;
LOCK 15.00;
SAVE;
EXIT;
```

LP46C



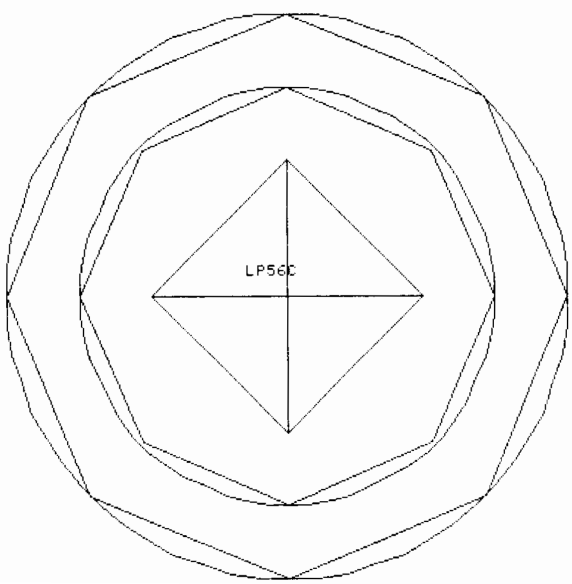

```
$FILES  LP46C $;
EDIT LP46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -50,-50  50,50;
ADD C27 :R6.00  0,0  35,0;
ADD C28 :R6.00  0,0  50,0;
ADD C29 :R45.00  0,0  35,0;
ADD C3 :R6.00  0,0  35,0;
ADD C30 :R45.00  0,0  50,0;
ADD C20 :R6.00  0,0  50,0;
ADD L1 :W0  -23,0  23,0;
ADD L1 :W0  0,-23  0,23;
ADD C1 .'D0046' :R90.00  0,0  23,0;
GRID 5,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

LP46S



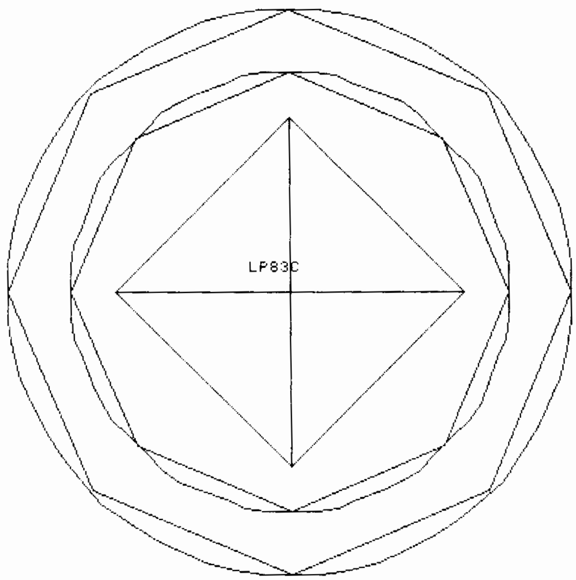

```
$FILES  LP46S $;
EDIT LP46S;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -50,-50  50,50;
ADD R29  -35,-35  35,35;
ADD R27  -35,-35  35,35;
ADD R30  -50,-50  50,50;
ADD R28  -50,-50  50,50;
ADD R20  -50,-50  50,50;
ADD L1 :W0  0,23  0,-23;
ADD L1 :W0  -23,0  23,0;
ADD C1 .'D0046' :R90.00  0,0  23,0;
ADD R3  -35,-35  35,35;
GRID 5,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

LP56C



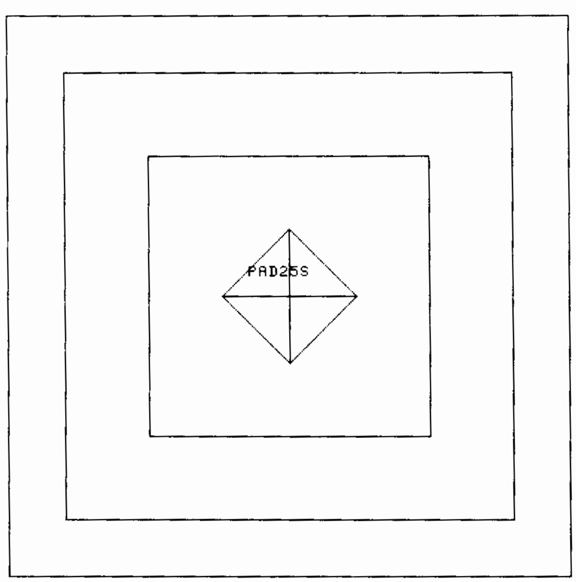

```
$FILES  LP56C $;
EDIT LP56C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -58,-58  58,58;
ADD C28 :R6.00  0,0  58,0;
ADD C30 :R45.00  0,0  58,0;
ADD C20 :R6.00  0,0  58,0;
ADD C27 :R6.00  0,0  43,0;
ADD C29 :R45.00  0,0  43,0;
ADD C3 :R6.00  0,0  43,0;
ADD L1 :W0  0,28  0,-28;
ADD L1 :W0  -28,0  28,0;
ADD C1 .'D0056' :R90.00  0,0  28,0;
GRID 1,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

LP83C



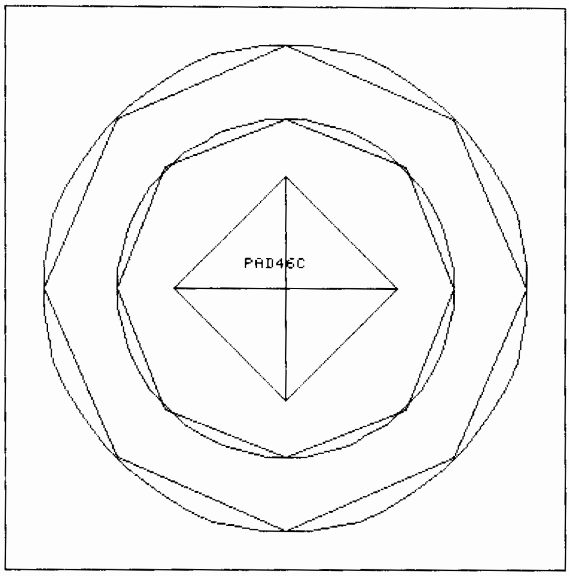

```
$FILES  LP83C $;
EDIT LP83C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -68,-68  68,68;
ADD C30 :R45.00  0,0  68,0;
ADD C28 :R6.00  0,0  68,0;
ADD C20 :R6.00  0,0  68,0;
ADD C29 :R45.00  0,0  53,0;
ADD C27 :R6.00  0,0  53,0;
ADD C3 :R6.00  0,0  53,0;
ADD L1 :W0  -42,0  42,0;
ADD L1 :W0  0,-42  0,42;
ADD C1 .´D0083´ :R90.00  0,0  42,0;
GRID 1,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

PAD25S



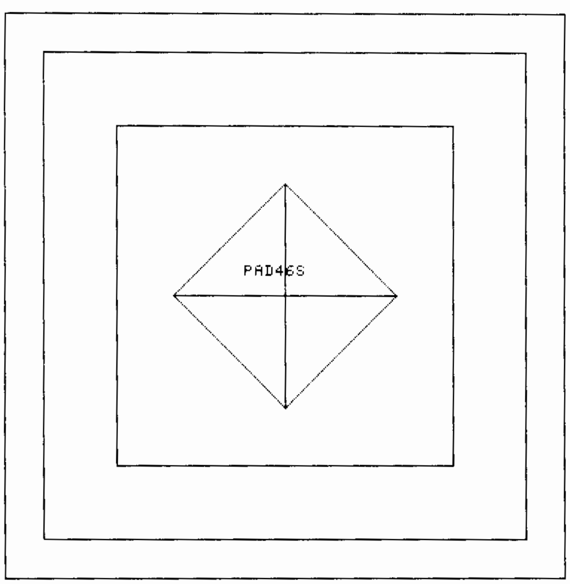

```
$FILES  PAD25S, LP25S $;
EDIT LP25S;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -40,-40  40,40;
ADD R28  -40,-40  40,40;
ADD R30  -40,-40  40,40;
ADD R20  -40,-40  40,40;
ADD R29  -25,-25  25,25;
ADD R27  -25,-25  25,25;
ADD L1 :W0  -12,0  12,0;
ADD L1 :W0  0,-12  0,12;
ADD C1 .´D0025´ :R90.00  0,0  12,0;
ADD R3  -25,-25  25,25;
GRID 5,1  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT PAD25S;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -50,-50  50,50;
ADD LP25S  0,0;
ADD R4  -50,-50  50,50;
GRID 1,5  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

PAD46C



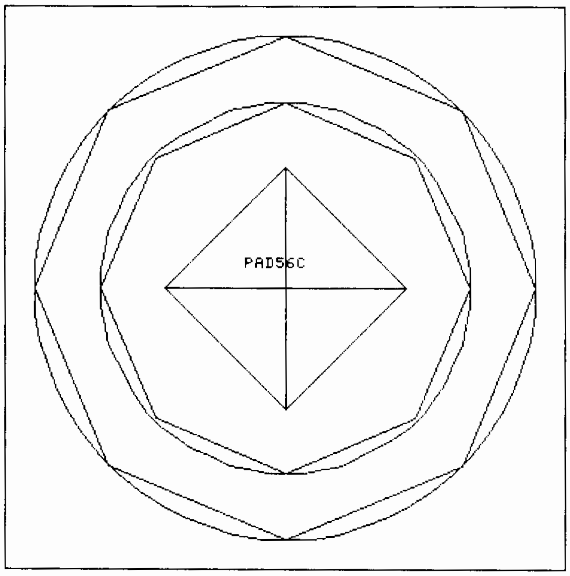

```
$FILES  PAD46C, LP46C $;
EDIT LP46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -50,-50  50,50;
ADD C27 :R6.00  0,0  35,0;
ADD C28 :R6.00  0,0  50,0;
ADD C29 :R45.00  0,0  35,0;
ADD C3 :R6.00  0,0  35,0;
ADD C30 :R45.00  0,0  50,0;
ADD C20 :R6.00  0,0  50,0;
ADD L1 :W0  -23,0  23,0;
ADD L1 :W0  0,-23  0,23;
ADD C1 .'D0046' :R90.00  0,0  23,0;
GRID 5,1  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT PAD46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -58,-58  58,58;
ADD LP46C  0,0;
ADD R4  -58,-58  58,58;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

PAD46S



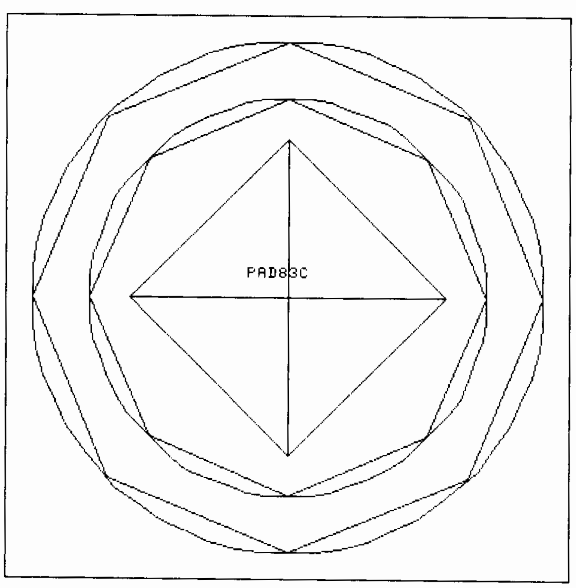

```
$FILES  PAD46S, LP46S $;
EDIT LP46S;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -50,-50  50,50;
ADD R29  -35,-35  35,35;
ADD R27  -35,-35  35,35;
ADD R30  -50,-50  50,50;
ADD R28  -50,-50  50,50;
ADD R20  -50,-50  50,50;
ADD L1 :W0  0,23  0,-23;
ADD L1 :W0  -23,0  23,0;
ADD C1 .'D0046' :R90.00  0,0  23,0;
ADD R3  -35,-35  35,35;
GRID 5,1  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT PAD46S;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -58,-58  58,58;
ADD LP46S  0,0;
ADD R4  -58,-58  58,58;
GRID 1,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

PAD56C



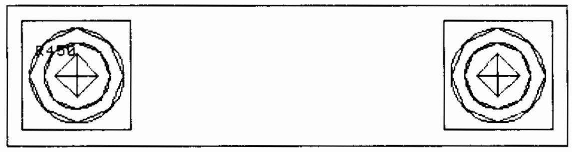

```
$FILES  PAD56C, LP56C $;
EDIT LP56C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -58,-58  58,58;
ADD C28 :R6.00  0,0  58,0;
ADD C30 :R45.00  0,0  58,0;
ADD C20 :R6.00  0,0  58,0;
ADD C27 :R6.00  0,0  43,0;
ADD C29 :R45.00  0,0  43,0;
ADD C3 :R6.00  0,0  43,0;
ADD L1 :W0  0,28  0,-28;
ADD L1 :W0  -28,0  28,0;
ADD C1 .'D0056' :R90.00  0,0  28,0;
GRID 1,1  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT PAD56C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -65,-65  65,65;
ADD LP56C  0,0;
ADD R4  -65,-65  65,65;
GRID 1,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

PAD83C



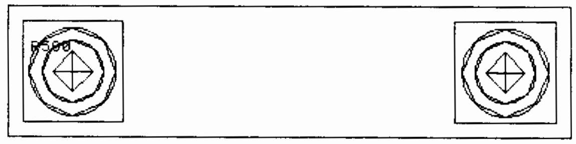

```
$FILES  PAD83C, LP83C $;
EDIT LP83C;
SHOW #E;
LOCK 0.0;
LEVEL 1;
GRID 1,1 0,0;
WINDOW  -68,-68  68,68;
ADD C30 :R45.00  0,0  68,0;
ADD C28 :R6.00  0,0  68,0;
ADD C20 :R6.00  0,0  68,0;
ADD C29 :R45.00  0,0  53,0;
ADD C27 :R6.00  0,0  53,0;
ADD C3 :R6.00  0,0  53,0;
ADD L1 :W0  -42,0  42,0;
ADD L1 :W0  0,-42  0,42;
ADD C1 .'D0083' :R90.00  0,0  42,0;
GRID 1,1  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT PAD83C;
SHOW #E;
LOCK 0.0;
LEVEL 1;
GRID 1,1 0,0;
WINDOW  -75,-75  75,75;
ADD LP83C  0,0;
ADD R4  -75,-75  75,75;
GRID 1,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

R450



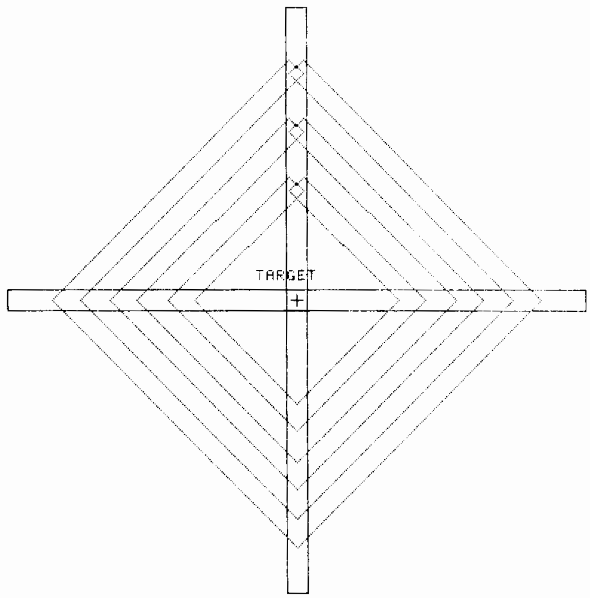

```
$FILES  R450, PAD46C, LP46C $;
EDIT LP46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -50,-50  50,50;
ADD C27 :R6.00  0,0  35,0;
ADD C28 :R6.00  0,0  50,0;
ADD C29 :R45.00  0,0  35,0;
ADD C3 :R6.00  0,0  35,0;
ADD C30 :R45.00  0,0  50,0;
ADD C20 :R6.00  0,0  50,0;
ADD L1 :W0  -23,0  23,0;
ADD L1 :W0  0,-23  0,23;
ADD C1 .'D0046' :R90.00  0,0  23,0;
GRID 5,1  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT PAD46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -58,-58  58,58;
ADD LP46C  0,0;
ADD R4  -58,-58  58,58;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT R450;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -75,-75  525,75;
ADD L26 :W0  525,75  525,-75;
ADD L26 :W0  -75,75  -75,-75;
ADD PAD46C  0,0;
ADD PAD46C  450,0;
ADD L26 :W0  -75,75  525,75;
ADD L26 :W0  -75,-75  525,-75;
GRID 5,5  0,0;
LOCK 15.00;
SAVE;
EXIT;
```

R500



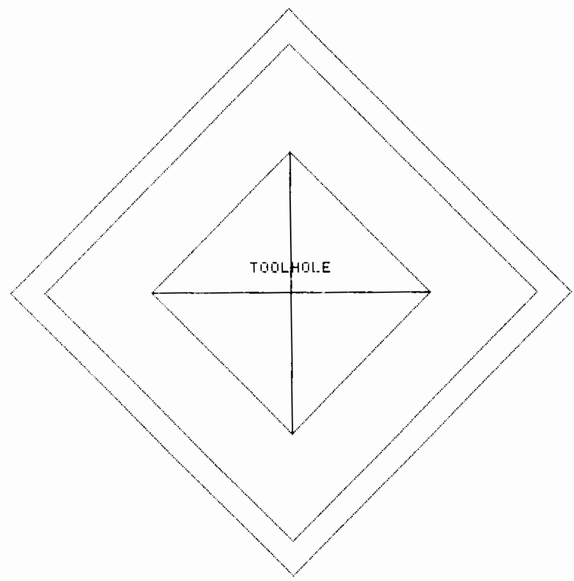

```
$FILES  R500, PAD46C, LP46C $;
EDIT LP46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -50,-50  50,50;
ADD C27 :R6.00  0,0  35,0;
ADD C28 :R6.00  0,0  50,0;
ADD C29 :R45.00  0,0  35,0;
ADD C3 :R6.00  0,0  35,0;
ADD C30 :R45.00  0,0  50,0;
ADD C20 :R6.00  0,0  50,0;
ADD L1 :W0  -23,0  23,0;
ADD L1 :W0  0,-23  0,23;
ADD C1 .'D0046' :R90.00  0,0  23,0;
GRID 5,1  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT PAD46C;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -58,-58  58,58;
ADD LP46C  0,0;
ADD R4  -58,-58  58,58;
GRID 1,10  0,0;
LOCK 0.00;
SAVE;
EXIT;

EDIT R500;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -75,-75  575,75;
ADD L26 :W0  575,75  575,-75;
ADD L26 :W0  -75,75  -75,-75;
ADD PAD46C  0,0;
ADD PAD46C  500,0;
ADD L26 :W0  -75,75  575,75;
ADD L26 :W0  -75,-75  575,-75;
GRID 5,5  0,0;
LOCK 15.00;
SAVE;
EXIT;
```

TARGET




```
$FILES  TARGET $;
EDIT TARGET;
SHOW #E;
LOCK 0.0;
GRID 1,1 0,0;
WINDOW  -170,-170  170,170;
ADD R19 .'ZTARGET'  -6,-6  6,6;
ADD L19 .'ZNULL' :W12  0,68  68,0  0,-68  -68,0  0,68;
ADD L19 .'ZNULL' :W12  0,102  102,0  0,-102  -102,0  0,102;
ADD L19 .'ZNULL' :W12  0,136  136,0  0,-136  -136,0  0,136;
ADD L19 .'ZNULL' :W12  0,170  0,-170;
ADD L19 .'ZNULL' :W12  170,0  -170,0;
GRID 17,1  0,0;
LOCK 0.00;
SAVE;
EXIT;
```

TOOLHOLE




```
$FILES  TOOLHOLE $;
EDIT TOOLHOLE;
SHOW #E;
LOCK 0.0;
LEVEL 1;
GRID 1,1 0,0;
WINDOW  -121,-121  121,121;
ADD C3 :R90.00  0,0  106,0;
ADD C2 :R90.00  0,0  106,0;
ADD C1 .'D0119' :R90.00  0,0  60,0;
ADD L1 :W0  0,-60  0,60;
ADD L1 :W0  -60,0  60,0;
ADD C20 :R90.00  0,0  121,0;
ADD C28 :R90.00  0,0  121,0;
ADD C30 :R90.00  0,0  121,0;
ADD C29 :R90.00  0,0  106,0;
ADD C27 .'ZDIAM' :R90.00  0,0  106,0;
GRID 1,10  0,0;
LOCK 15.00;
SAVE;
EXIT;
```

Table of Contents

Syntax Reference

Information Provided

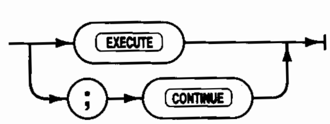
This section contains an alphabetical reference to all the commands currently available with EGS-45. Each entry defines the command, shows the proper syntax for its use, provides some examples of the command's use, and explains the details of using the command. Above each syntax diagram is a small table indicating in which modules/subsystems the command is valid.

Syntax Diagrams Explained

Command syntax is represented pictorially. All characters enclosed by a rounded envelope must be entered exactly as shown. Words enclosed by a rectangular box are names of items used with the command. A description of each item is given in the table following the drawing. Statement elements are connected by lines. Each line can be followed in only one direction, as indicated by the arrow at the end of the line. Any combination of command elements that can be generated by following the lines in the proper direction is syntactically correct. An element is optional if there is a valid path around it. Optional items often have default values. The table or text following the drawing specifies the default value that is used when an optional item is not included in a command.

Some items in the diagrams are not enclosed by a rectangle or an envelope, but instead are enclosed in parentheses. These items represent macros supplied in the macro file, MACRDATA supplied with your EGS-45 software. When this macro file is loaded, you may enter the macro instead of the item associated with the macro in the diagram. For example, you may enter $ROTATE instead of :R when adding a note component with the ADD command.

Commands are usually terminated by entering ;. When typing a command from the computer keyboard, you may either type ; and press the CONTINUE key or simply press the EXECUTE key. This is indicated with the diagram below.



For commands entered in the edit subsystem, selecting ; from the menu area of the display is equivalent to entering the sequence of keys indicated above.

The diagrams do not deal with the proper use of spaces and commas. The computer uses spaces and commas to distinguish the boundaries between various commands, parameters, data, and other items. The following are some general rules for using spaces and commas when entering a command.

Between a Command Name and Options

A space or comma is used to separate a key item (one enclosed in a rounded envelope) from a preceding key item or data associated with a previous key item. For example, with the ADD command a space (or comma) is used to separate the descriptor from the command name:

```
ADD L2 ...
```

Similarly, a space or comma separates the :R parameter from data immediately preceding it in the ADD command:

```
ADD T2 :R45 ...
```

In the edit subsystem, if commands and parameters are selected from the menu area of the display, it is not necessary to enter a space or comma between selected items.

Between Parameters and Data

No space or comma should be entered between a key item and associated data. For example, when entering the ADD command, no space should separate the descriptor and the layer number:

```
ADD A1 ...
```

In the edit subsystem, if commands and data are selected from the menu area of the display, it is not necessary to enter a space or comma between selected items. However, when selecting numeric values from the numeric column of the display, it is necessary to separate individual values by selecting a space or comma.

Note

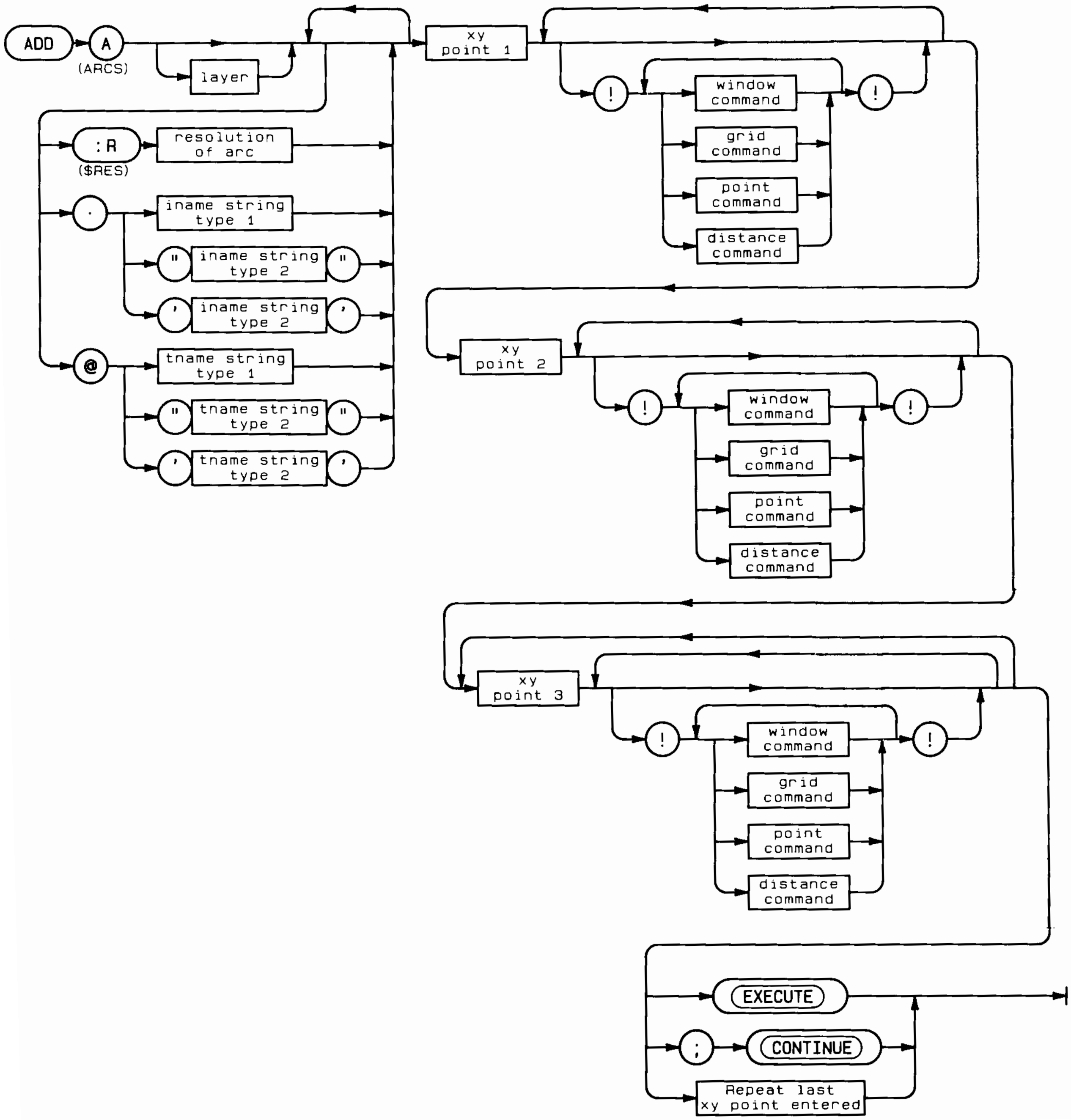
When using macros to replace a descriptor, a space must separate the macro name and any associated data. For example, using the macro ARCS (supplied with EGS-45 in the macro file MACRDATA):

```
ADD ARCS 1 ...                MOVE ARCS x,y ...
```

ADD A (ADD ARCS)

Valid in:
Edit subsystem

ADD A allows an arc component to be added to the drawing.



Item	Description/Default	Range Restrictions
A	A descriptor specifying an arc component.	
layer	Layer in which you wish to add the arc. If the layer is not specified, the system defaults to the last layer number entered, and displays that number at the top of the screen.	1 - 255
resolution of arc	Specifies the angle (in degrees) between endpoints of a line segments approximating the arc. The default resolution of the arc is equal to the current lock angle (see the LOCK command for more information.	minimum value = 6 maximum value = 120
iname string type 1	Set of characters forming the iname that you wish to assign to the arc. All characters may be displayed and plotted. The first character of the string must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
iname string type 2	Set of characters forming the iname that you wish to assign to the arc. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
tname string type 1	Set of characters forming the tname that you wish to assign to the arc. All characters may be displayed and plotted. The first character of the string must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
iname string type 2	Set of characters forming the tname that you wish to assign to the arc. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (>), (<), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
xy point 1	One endpoint of the arc.	
!	Specifies to interrupt (suspend) the ADD command. Once the command is suspended, you may enter any of the following commands (complete with any of the legal options): the WINDOW command, the GRID command, the POINT command, or the DISTANCE command. This allows you to change the location of the window, change the current user or display grid, or obtain information (such as the xy value of a particular location with the POINT command). Re-entering ! allows you to continue entering the ADD command from the point at which it was interrupted.	
xy point 2	The other endpoint of the arc.	

xy point 3	A point lying on the arc between xy point 1 and xy point 2. The resulting arc is displayed on the CRT. This point is called a trial placement location because entering a new xy point 3 causes the old value of xy point 3 to be replaced with the new value. The result is displayed on the CRT.	A maximum of 255 trial placement locations may be entered.

HELP syntax

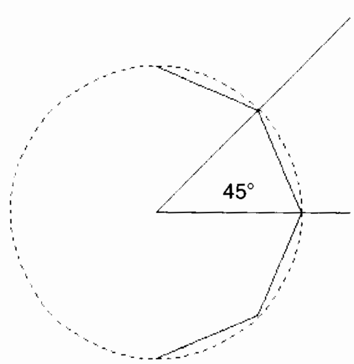
```
ADD {desc[layer]|[I] devicename} [.iname][@tname] xy1 [xy2 ...] EOC
```

Use

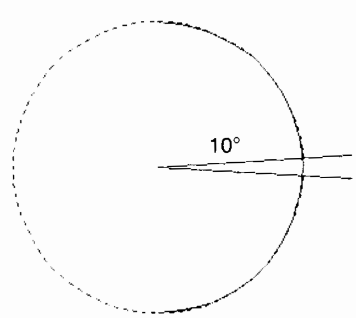
Entering the ADD command with the arc descriptor allows an arc component to be added to the drawing. The arc is approximated by a section of a regular polygon (a polygon whose sides are all of equal length). It is defined by entering two endpoints of the arc and a third point which lies on the arc (see the drawing below). The system then computes the size of the regular polygon needed to pass through all three points and plots it in the display.

The interrupt option allows you to change the position of the window, change the current user or display grid, or obtain information. For example, if you find that the window is too small or in the wrong position to add the arc component, you may suspend the ADD command and reposition the window. Then, by entering ! again, you may continue to add the arc to your drawing by completing the ADD command from the point at which it was suspended.

By specifying the resolution of the arc you can determine the smoothness of the curve of the arc and the speed with which it is drawn. The resolution of the arc is a measure of the angle (in degrees) between endpoints of the line segments approximating the arc (see the drawing below). As the resolution of the arc becomes finer, it takes greater amounts of time for the system to draw the arc.



Resolution = 45°



Resolution = 10°

Example

```
ADD A3 :R6 ."25.13%" 0,0  0,50  20,25 ;
```

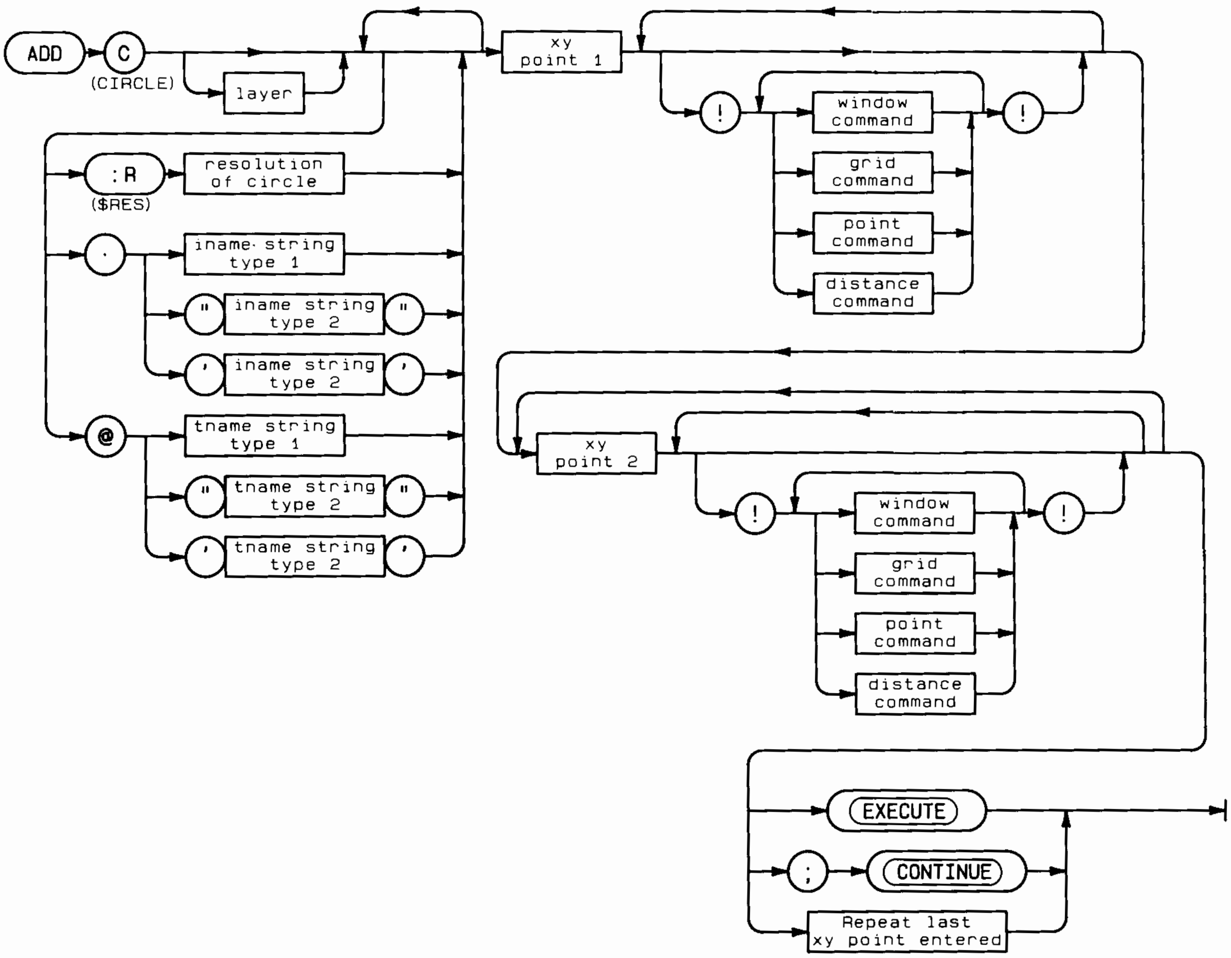
result:

An arc is added to layer 3 in the current drawing. The arc has endpoints of 0,0 and 0,50 and the point 20,25 lies on the circular arc generated. The arc has a resolution of 6 degrees. The iname, 25.13%, is assigned to the arc.

ADD C (ADD CIRCLE)

Valid in:
Edit subsystem

The ADD command with a circle descriptor allows a circle component to be added to the drawing.



Item	Description/Default	Range Restrictions
C	Descriptor specifying a circle component.	
layer	Layer to which you wish to add a circle. If the layer is not specified, the system defaults to the last layer number entered, and displays that number at the top of the screen.	1 - 255
resolution of circle	Specifies the angle (in degrees) between endpoints of the line segments approximating the circle. The default resolution of the circle is equal to the current lock angle (see the LOCK command for more information.	minimum = 6 maximum = 120
iname string type 1	Set of characters forming the iname that you wish to assign to the circle. All characters may be displayed and plotted. The first character of the string must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
iname string type 2	Set of characters forming the iname that you wish to assign to the circle. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]) (_), ($), (?), (=)
tname string type 1	Set of characters forming the tname that you wish to assign to the circle. All characters may be displayed and plotted. The first character of the string must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
tname string type 2	Set of characters forming the iname that you wish to assign to the circle. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9,(<), (>), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
xy point 1	xy point locating the center of the circle to be added.	
!	Specifies to interrupt (suspend) the ADD command. Once the command is suspended, you may enter any of the following commands (complete with any of the legal options): the WINDOW command, the GRID command, the POINT command, or the DISTANCE command. This allows you to change the location of the window, change the current user of display grid, or obtain information (such as the xy value of a particular location with the POINT command). Re-entering ! allows you to continue entering the ADD command from the point at which it was interrupted.	
xy point 2	xy point lying on the circumference of the circle. The resulting circle is displayed on the drawing. This point is called a trial placement location because entering a new xy point 2 causes the old value of xy point 2 to be replaced with the new value. The result is displayed on the CRT.	a maximum of 254 trial placement locations may be entered.

HELP syntax

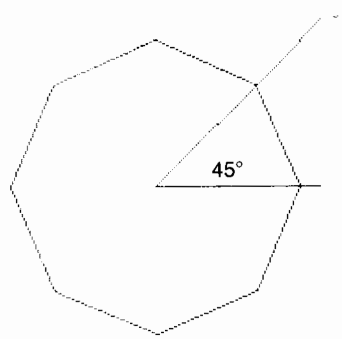
```
ADD {desc[layer]|[I] devicename} [.iname][@tname] xy1 [xy2 ...] EOC
```

Use

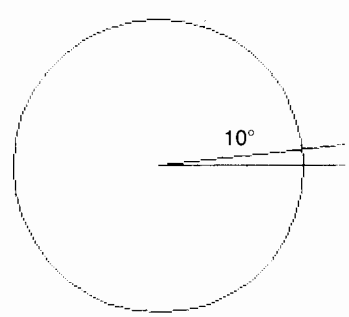
Entering the ADD command with the circle descriptor allows a circle component to be added to the drawing. The circle is approximated by a regular polygon (a polygon whose sides are all of equal length). It is defined by the center of the circle and a point which lies on the circle (see the drawing below).

The interrupt option allows you to change the position of the window, change the current user or display grid, or obtain information. For example, if you find that the window is too small or in the wrong position to add the circle component, you may suspend the ADD command and reposition the window. Then, by entering ! again, you may continue to add the circle to your drawing by completing the ADD command from the point at which it was suspended.

By specifying the resolution of the circle you can determine the smoothness of the curve of the circle and the speed with which it is drawn. The resolution of the circle is a measure of the angle (in degrees) between endpoints of the line segments approximating the circle (see the drawing below). As the resolution of the circle becomes finer, it takes greater amounts of time to draw the circle.



Resolution = 45°



Resolution = 10°

Example

```
ADD C2 0,0  25,0  35,0 ;
```

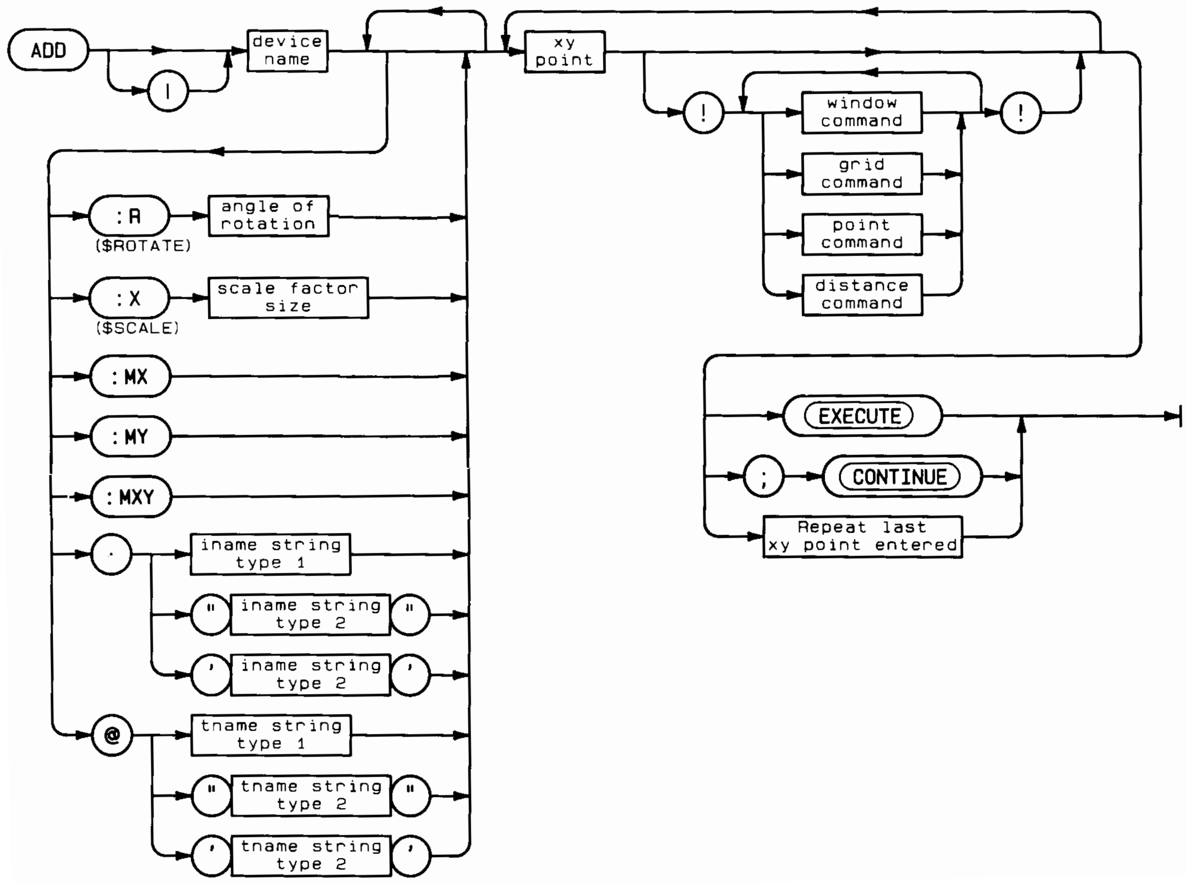
result:

A circle is added to layer 2 of the current drawing. The resolution of this circle is equal to the current lock angle of the drawing, since no resolution is specified. First a circle is displayed with its center at the xy point 0,0 and the point 25,0 lying on the circumference of the circle (i.e., radius = 25). Entering the third xy point 35,0 causes the original circle to be replaced by a circle with its center at 0,0 and the point 35,0 on its circumference (i.e., radius = 35).

ADD I

Valid in:
Edit subsystem

The ADD command with an optional descriptor and a device name (name of the device to which you wish to link your drawing), allows an instance component to be added to your drawing.



Item	Description/Default	Range Restrictions
I	Descriptor specifying an instance component.	
device name	Name of the device which you wish to link to your drawing.	
angle of rotation	Angle, in degrees, which you wish to rotate the instance component, when added to the drawing. The components of the linked device are displayed at the rotated angle.	−180.00 to 180.00
scale factor	Amount by which you wish to change the scale of the existing device when displayed with your drawing. The current units of the device is multiplied by the scale factor to obtain the new scale. For example, entering a scale factor of .5 causes the dimensions of the device linked to your drawing to be one half of the original dimensions of the device. Entering a negative scale factor causes the scale of the device linked to your drawing to be divided by the scale factor, instead of multiplied. For example, entering the scale factor −2, causes the dimensions of the linked device to be one half of the original dimensions.	
:MX	Specifies that the instance component added to the drawing is to be mirrored about the X axis (see the drawings below). The components forming the linked device are mirrored in the same pattern, when displayed.	
:MY	Specifies that the instance component added to the drawing is to be mirrored about the Y axis (see the drawings below). The components forming the linked device are mirrored in the same pattern, when displayed.	
:MXY	Specifies that the instance component add to the drawing is to be mirrored about both the X and the Y axes. The components forming the linked device are mirrored in the same pattern, when displayed.	
iname string type 1	Set of characters forming the iname that you wish to assign to the instance. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
iname string type 2	Set of characters forming the iname that you wish to assign to the instance. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)

tname string type 1	Set of characters forming the tname that you wish to assign to the instance. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
tname string type 2	Set of characters forming the iname that you wish to assign to the instance. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (@), (^), ([), (]), (_), ($), (?), (=)
xy point	Specifies the xy location on the drawing where you wish to add the instance component. The instance is added such that the origin of the instance is placed exactly at the xy point. This point is called a trial placement location because entering a new xy point causes the instance to be moved from the old location to the new location. The result is displayed on the CRT.	a maximum of 255 trial placement locations may be entered.
!	Specifies to interrupt (suspend) the ADD command. Once the command is suspended, you may enter any of the following commands (complete with any of the legal options): the WINDOW command, the GRID command, the POINT command, or the DISTANCE command. This allows you to change the location of the window, change the current user of display grid, or obtain information (such as the xy value of a particular location with the POINT command). Re-entering ! allows you to continue entering the ADD command from the point at which it was interrupted.	

HELP syntax

```
ADD {desc[layer]|[I] devicename} [.iname][@tname] xy1 [xy2 ...] EOC
```

Use

Enter the ADD command along with the name of the device which you wish to link to your drawing. If no device exists with that name, the system assumes that the device will be created later, places a dot along with the device name at the location where the device will be displayed when created.

If the current user units (as defined in the process file) do not match the units of the linked device, the system automatically converts the units of the device to match the current user units. This is done before any of the command options (including the scale option) are executed. The system issues a message before converting, which tells you that the units of the linked device do not match the current user units. The system then converts the units of the linked device.

For example, suppose that the units of the linked device are one inch per system grid point (`UNITS IN,1`). If the current user units are one foot per system grid point, the system converts the units of the device to one foot per system grid point as the device is linked. Thus if a component in the device was originally 1 inch long, it is 1 foot long when displayed with your current drawing.

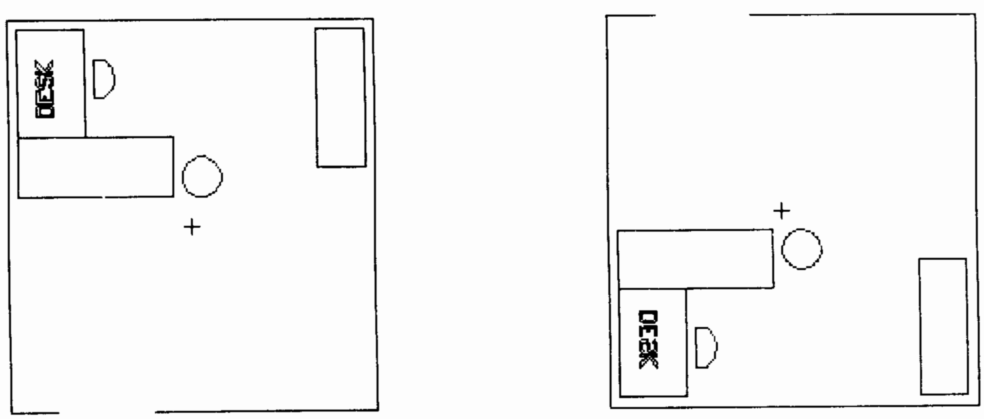
The interrupt option allows you to change the position of the window, change the current user or display grid, or obtain information (via the DISTANCE or POINT commands). For example, suppose that while attempting to place the instance component in your drawing, you find that the window is in the wrong location to allow proper placement of the component. You may suspend the ADD command and reposition the window with the WINDOW command. Then, by entering ! again, you may continue to add the instance component to your drawing by completing the command from the point at which it was suspended.

The instance may be rotated about its origin and may be scaled to fit the dimensions of your drawing. Additionally, the instance may be mirrored about the X and/or Y axes.

Example

```
ADD EXAMP  :X2  :MX  15,15 ;
```

result:



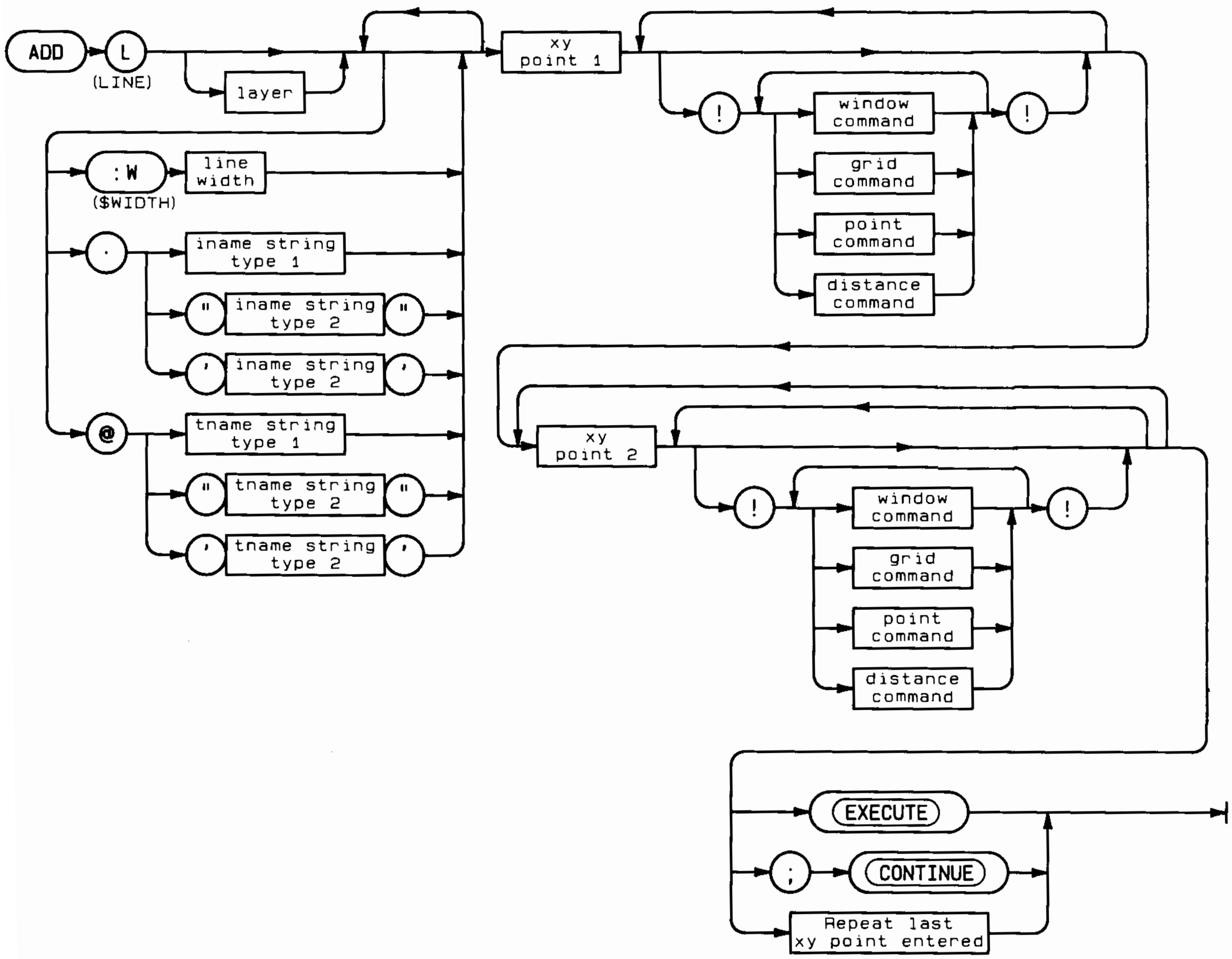
Device as stored　　Instance as added

An instance of the device EXAMP is added to the drawing, such that its origin is placed at the location 15,15. The scale of the instance that is added is twice that of the original. The instance that is added is mirrored about the X axis.

ADD L (ADD LINE)

Valid in:
Edit subsystem

The ADD command with a line descriptor allows a line to be added to the drawing. The line may be segmented with up to 255 vertices (including endpoints). The line may have a width, which is displayed symetrically about the center of the line.



Item	Description/Default	Range Restrictions
layer	Layer number in which you wish to add the line. If the layer is not specified, the system defaults to the last layer number entered, and displays that number at the top of the screen.	1 - 255
line width	Width of the line, in current user units. The default line width is 0.	width > 0 maximum width is the number of user units assigned to 16383 system grid points.
iname string type 1	Set of characters forming the iname that you wish to assign to the line. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
iname string type 2	Set of characters forming the iname that you wish to assign to the line. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
tname string type 1	Set of characters forming the tname that you wish to assign to the line. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
tname string type 2	Set of characters forming the iname that you wish to assign to the line. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
xy point 1	Location of the beginning of the line, an endpoint.	
!	Specifies to interrupt (suspend) the ADD command. Once the command is suspended, you may enter any of the following commands (complete with any of the legal options): the WINDOW command, the GRID command, the POINT command, or the DISTANCE command. This allows you to change the location of the window, change the current user or display grid, or obtain information (such as the xy value of a particular location with the POINT command). Re-entering ! allows you to continue entering the ADD command from the point at which it was interrupted.	
xy point 2	Location of a vertex in the segmented line, or the other endpoint of the line.	up to 255 points may be entered

HELP syntax

```
ADD {desc[layer]|[I] devicename} [.iname][@tname] xy1 [xy2 ...] EOC
```

Use

The ADD command with a line descriptor allows a line to be added to the drawing. The line may be segmented with up to 253 vertices (plus the two endpoints). The line may have a width, which is displayed symetrically about the center of the line. The line is defined by entering an endpoint, any vertices (junction of line segments), and finally, a second endpoint.

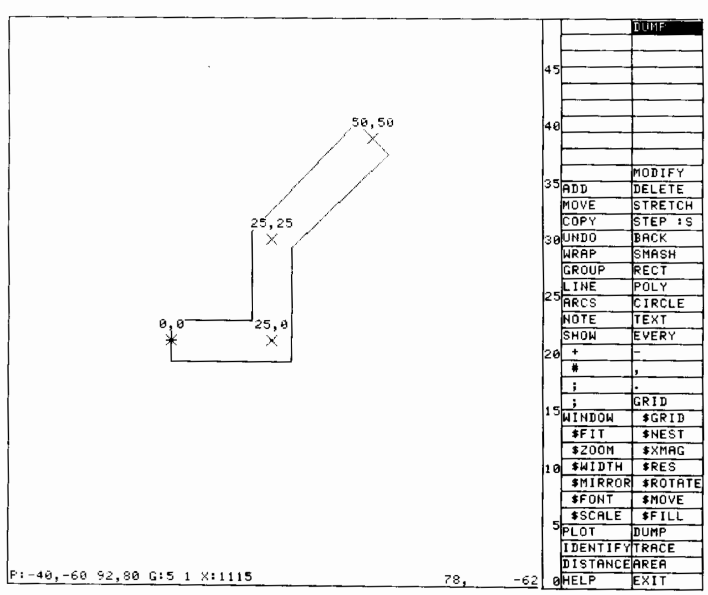
The interrupt option allows you to change the position of the window, change the current user or display grid, or obtain information. For example, suppose you are adding a line to your drawing and find that one end of the line should extend to a portion of the drawing not displayed in the window. You can suspend the ADD command by entering !. Then using the WINDOW command, you can reposition the window so that the window displays the location for the end of the line component. By re-entering !, you may then complete the line by continuing the ADD command from the point that it was suspended.

Example

```
ADD L1 :W10 .CC10 0,0 25,0 25,25 50,50;
```


result:

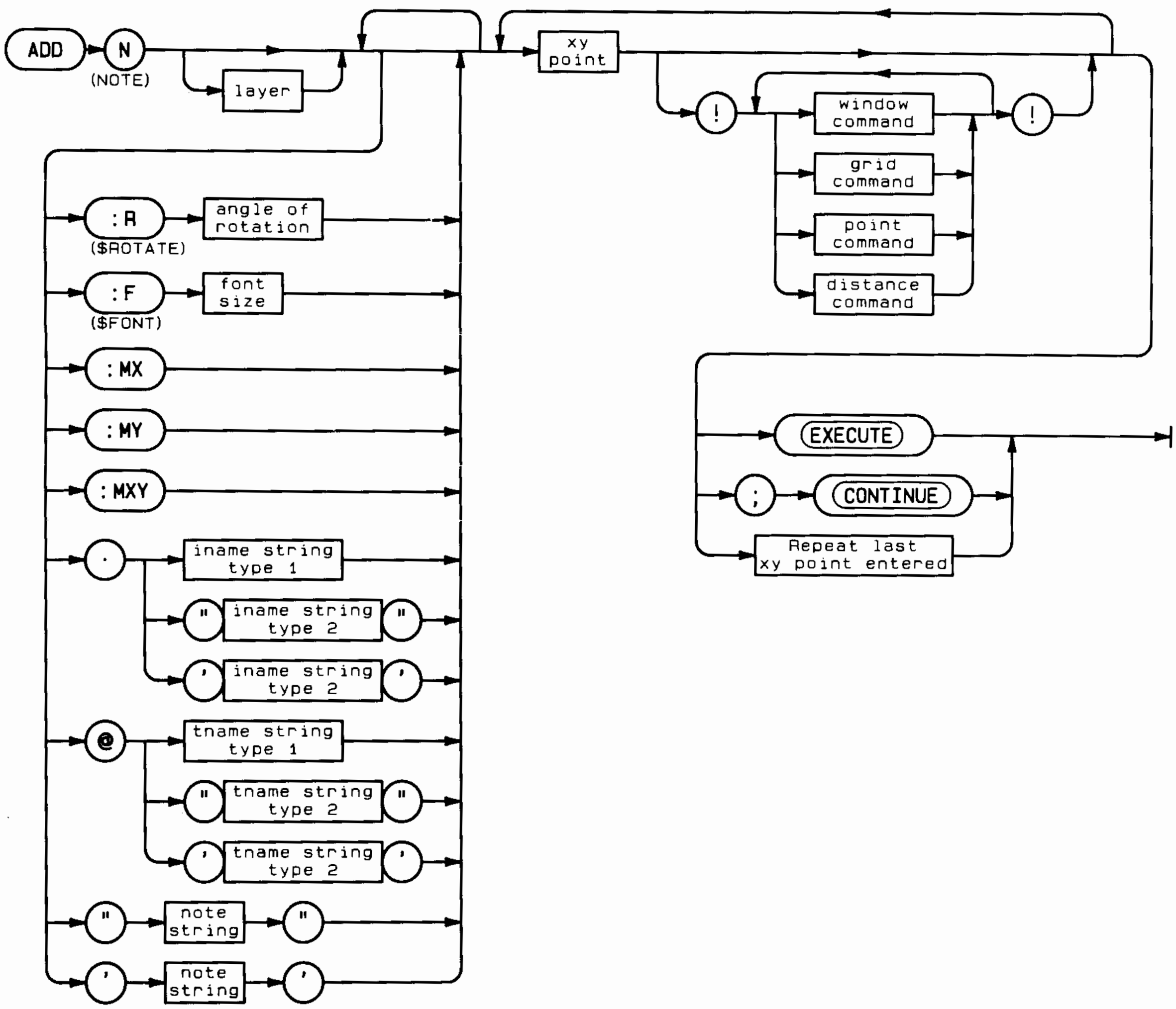
A segmented line with width 10 is added to layer 1 in the current drawing beginning at 0,0 and ending at 50,50. The line has vertices at 25,0 and at 25,25 and is assigned the iname, CC10.



ADD N (ADD NOTE)

Valid in:
Edit subsystem

The ADD command with a note descriptor allows a note component to be added to the drawing.



Note: the string must be specified at least once before attempting to enter an xy point

Item	Description/Default	Range Restrictions
layer	Layer number in which you wish to add the note. If the layer is not specified, the system defaults to the last layer number entered, and displays that number at the top of the screen.	1 - 255
angle of rotation	Angle, in degrees, at which you wish to rotate the note being added to the drawing. The default is 0 degrees.	−180.00 to 180.00
font size	Height of the characters, in user units, from which the note is to be formed. The default font size is 10 user units. The maximum font size is determined by the total distance, in user units, between 16383 system grid points.	1 - (number of user units assigned to 16383 system grid points)

:MX	Specifies that the note is to be mirrored about the X axis, when added to the drawing.	
:MY	Specifies that the note is to be mirrored about the Y axis, when added to the drawing.	
:MXY	Specifies that the note is to be mirrored about both the X and the Y axes, when added to the drawing.	
iname string type 1	Set of characters forming the iname that you wish to assign to the note. All characters may be displayed and plotted.	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
iname string type 2	Set of characters forming the iname that you wish to assign to the note. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
tname string type 1	Set of characters forming the tname that you wish to assign to the note. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
tname string type 2	Set of characters forming the iname that you wish to assign to the note. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
note string	Set of characters forming the note that you wish to add to the drawing.	64 characters maximum. legal characters: A-Z, 0-9, (_), (*), (<), (>), ([), (]), (.), (,), (+), (−), (/), (\), (∧)
xy point	Location on the drawing at which you wish to add the note. The note is added such that the lower left corner of the note is placed at the xy point, and then is rotated about this point if rotation is specified. This point is called a trial placement location because entering a new xy point causes the note to be moved from the previous location to the new location. The result is displayed on the CRT.	maximum of 255 trial placement points may be entered.
!	Specifies to interrupt (suspend) the ADD command. Once the command is suspended, you may enter any of the following commands (complete with any of the legal options): the WINDOW command, the GRID command, the POINT command, or the DISTANCE command. This allows you to change the location of the window, change the current user or display grid, or obtain information (such as the xy value of a particular location with the POINT command). Re-entering ! allows you to continue entering the ADD command from the point at which it was interrupted.	maximum of 255 trial placement points may be entered.

HELP syntax

```
ADD {desc[layer]|[I] devicename} [.iname][@tname] xy1 [xy2 ...] EOC
```

Use

The ADD command with a note descriptor allows a note component to be added at the xy location specified. Entering additional xy points causes the note to be shifted to the new location(s). Thus, the location specified with xy point is a trial placement point.

The note is a "stick letter" representation of the characters supplied. It may be rotated about its origin, which is located at the lower left hand corner of the note. The size of the characters forming the note may also be specified. The note may be mirrored about the X and/or Y axes.

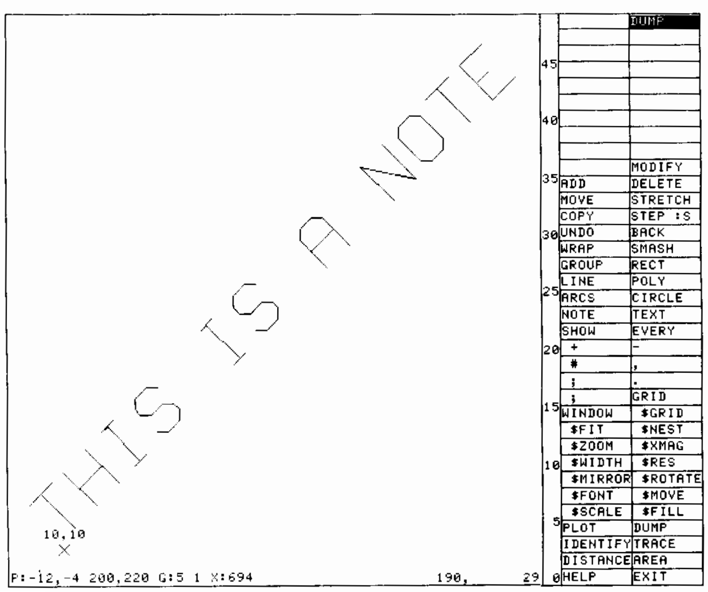
The interrupt option allows you to change the position of the window, change the current user or display grid, or obtain information. For example, if you find that the window is too small or is in the wrong position to add the note, you may simply suspend the ADD command and reposition the window with the WINDOW command. Then, by entering ! again, you may continue to add the note component to your drawing by completing the command from the point at which it was suspended.

Example

```
ADD N2 :R45 :F25 "THIS IS A NOTE"  10,10 ;
```

result:

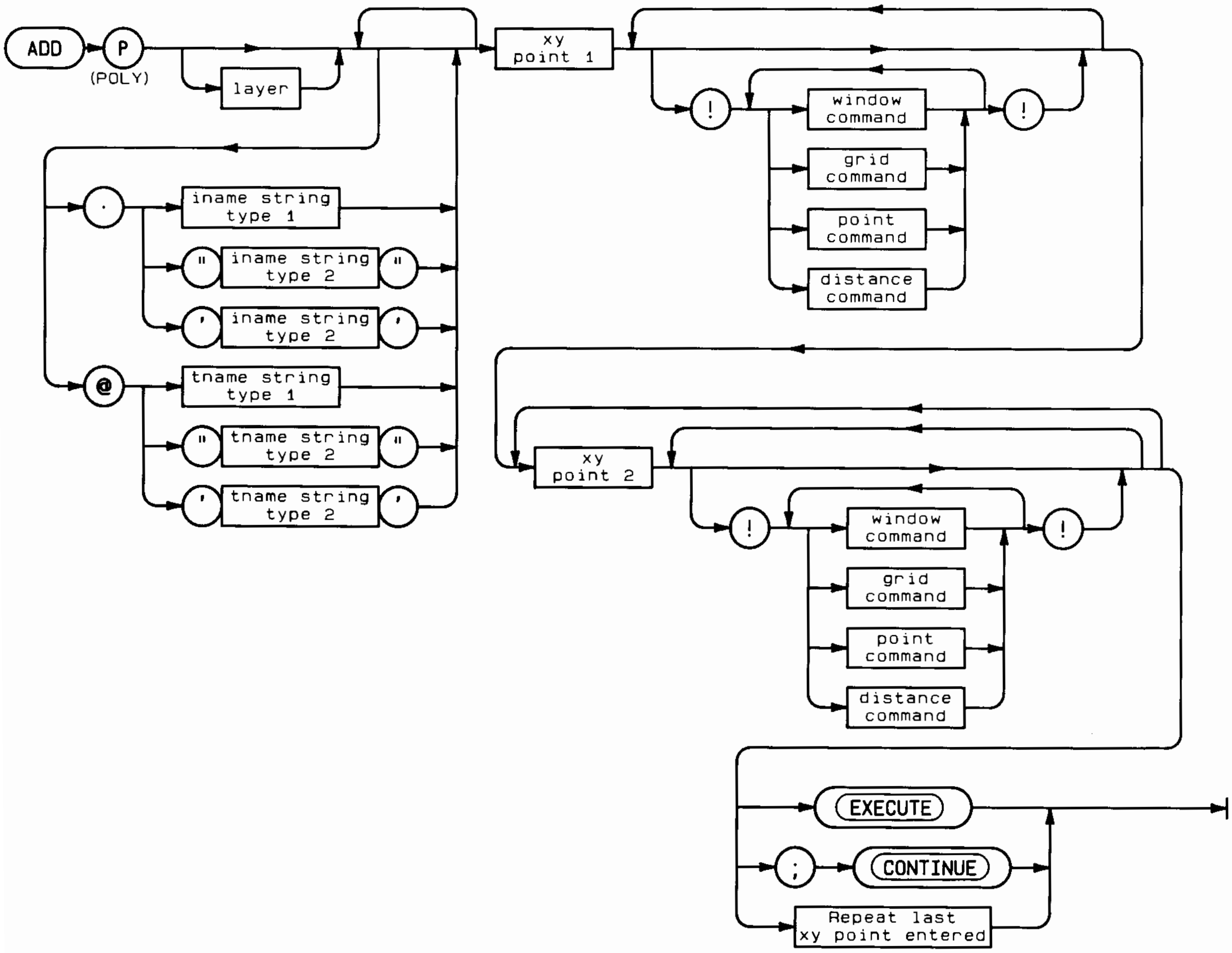
a note is added to layer 2 with characters 25 units tall. The note is rotated 45 degrees about its origin and is placed at the location 10,10.



ADD P (ADD POLY)

Valid in:
Edit subsystem

The ADD command with a polygon descriptor allows a polygon component to be added to the drawing.



Item	Description/Default	Range Restrictions
layer	Layer number in which you wish to add the polygon. If the layer is not specified, the system defaults to the last layer number entered, and displays that number at the top of the screen.	1 - 255
iname string type 1	Set of characters forming the iname that you wish to assign to the polygon. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
iname string type 2	Set of characters forming the iname that you wish to assign to the polygon. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (%), (/), (\), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
tname string type 1	Set of characters forming the tname that you wish to assign to the polygon. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
tname string type 2	Set of characters forming the iname that you wish to assign to the polygon. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (%), (/), (\), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
xy point 1	Location of one vertex of the polygon.	
!	Specifies to interrupt (suspend) the ADD command. Once the command is suspended, you may enter any of the following commands (complete with any of the legal options: the WINDOW command, the GRID command, the POINT command, or the DISTANCE command. This allows you to change the location of the window, change the current user or display grid, or obtain information (such as the xy value of a particular location via the POINT command). Re-entering ! allows you to continue entering the ADD command from the point at which it was interrupted.	
xy point 2	Location of the second vertex of the polygon, moving either clockwise or counter clockwise around the polygon.	
xy point 3	Location of the third vertex of the polygon. Entering additional xy points specifies additional vertices of the polygon. A maximum of 255 verticies may be entered.	

HELP syntax

```
ADD {desc[layer]|[I] devicename} [.iname][@tname] xy1 [xy2 ...] EOC
```

Use

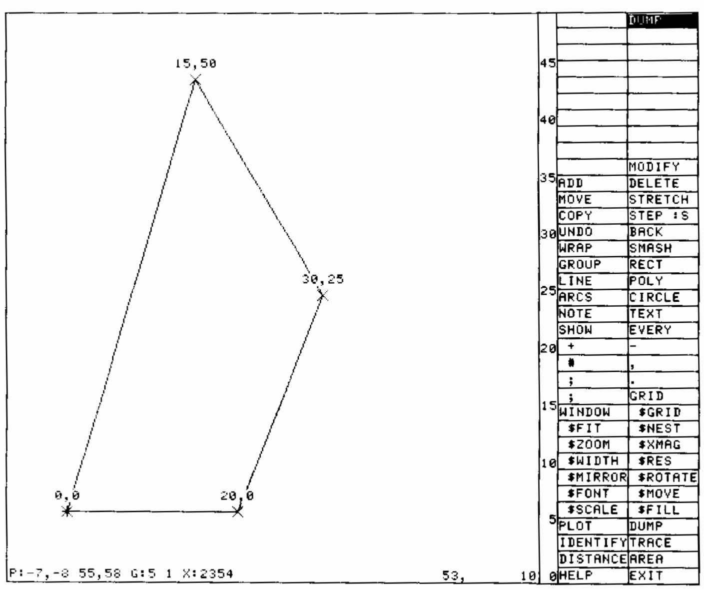
Entering the ADD command with a polygon descriptor allows you to add a polygon component to the drawing by specifying xy points that represent the location of the vertices of the polygon. The polygon generated must have at least three sides.

The interrupt option allows you to change the position of the window, change the current user or display grid, or obtain information. For example, if you find that the window is too small or is in the wrong position to add the polygon component, you may suspend the ADD command and then use the WINDOW command to reposition the window. Then, by entering ! again, you may continue to add the polygon component to your drawing by completing the command from the point at which it was suspended.

Example

```
ADD P3 0,0  20,0  30,25  15,50 ;
```

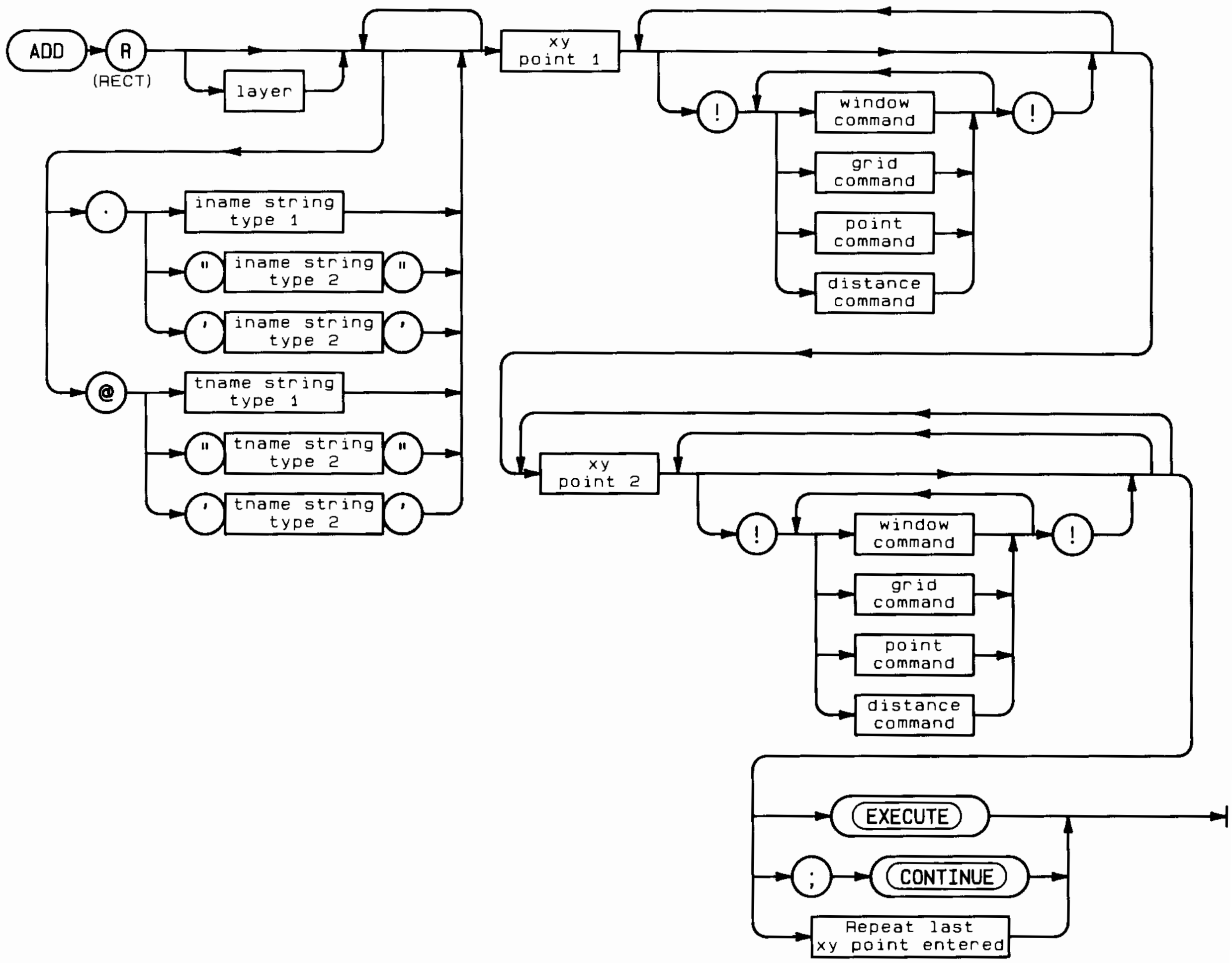
result:



ADD R (ADD RECT)

Valid in:
Edit subsystem

The ADD command with a rectangle descriptor allows a rectangle component to be added to the drawing.



Item	Description/Default	Range Restrictions
layer	Layer number in which you wish to add the polygon. If the layer is not specified, the system defaults to the last layer number entered, and displays that number at the top of the screen.	1 - 255
iname string type 1	Set of characters forming the iname that you wish to assign to the polygon. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
iname string type 2	Set of characters forming the iname that you wish to assign to the polygon. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (%), (/), (\), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
tname string type 1	Set of characters forming the tname that you wish to assign to the polygon. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
tname string type 2	Set of characters forming the iname that you wish to assign to the polygon. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (%), (/), (\), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
xy point 1	Location of one corner of the rectangle.	
!	Specifies to interrupt (suspend) the ADD command. Once the command is suspended, you may enter any of the following commands (complete with any of the legal options: the WINDOW command, the GRID command, the POINT command, or the DISTANCE command. This allows you to change the location of the window, change the current user or display grid, or obtain information (such as the xy value of a particular location via the POINT command). Re-entering ! allows you to continue entering the ADD command from the point at which it was interrupted.	
xy point 2	Location of the corner of the rectangle opposite xy point 1. This point is called a trial placement location because entering a new xy point2 causes the old value of xy point2 to be replaced with the new value. The result is displayed on the CRT.	a maximum of 255 trial placement locations may be entered.

HELP syntax

```
ADD {desc[layer]|[I] devicename} [.iname][@tname] xy1 [xy2 ...] EOC
```

Use

Entering the ADD command with a rectangle descriptor allows a rectangle component to be added to the drawing. The rectangle is defined by the two endpoints of one of the diagonals of the rectangle. The location of xy point 1 marks one endpoint of the diagonal, while xy point 2 marks the other endpoint. Entering additional xy points causes the system to replace the previous value of xy point 2 with the new xy location(s).

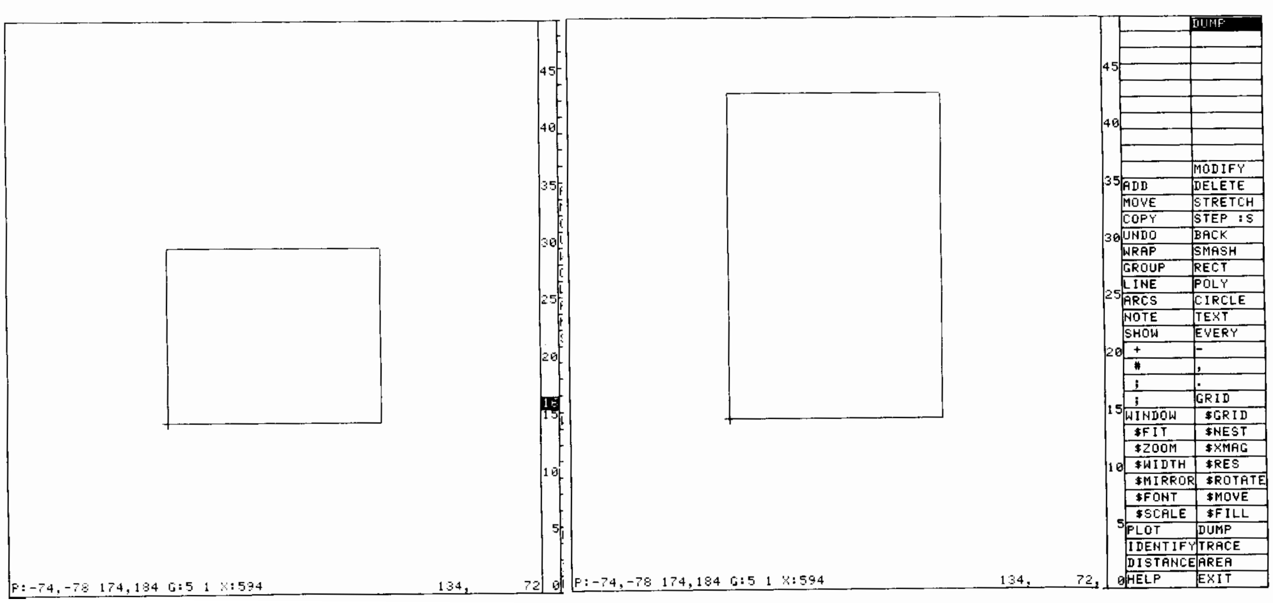
The interrupt option allows you to change the position of the window, change the current user or display grid, or obtain information. For example, if you find that the window is too small or is in the wrong position to add the rectangle component, you may suspend the ADD command and then use the WINDOW command to reposition the window. Then, by entering ! again, you may continue to add the rectangle component to your drawing by completing the command from the point at which it was suspended.

Example:

```
ADD R2  0,0  100,80  100,150;
```

result:

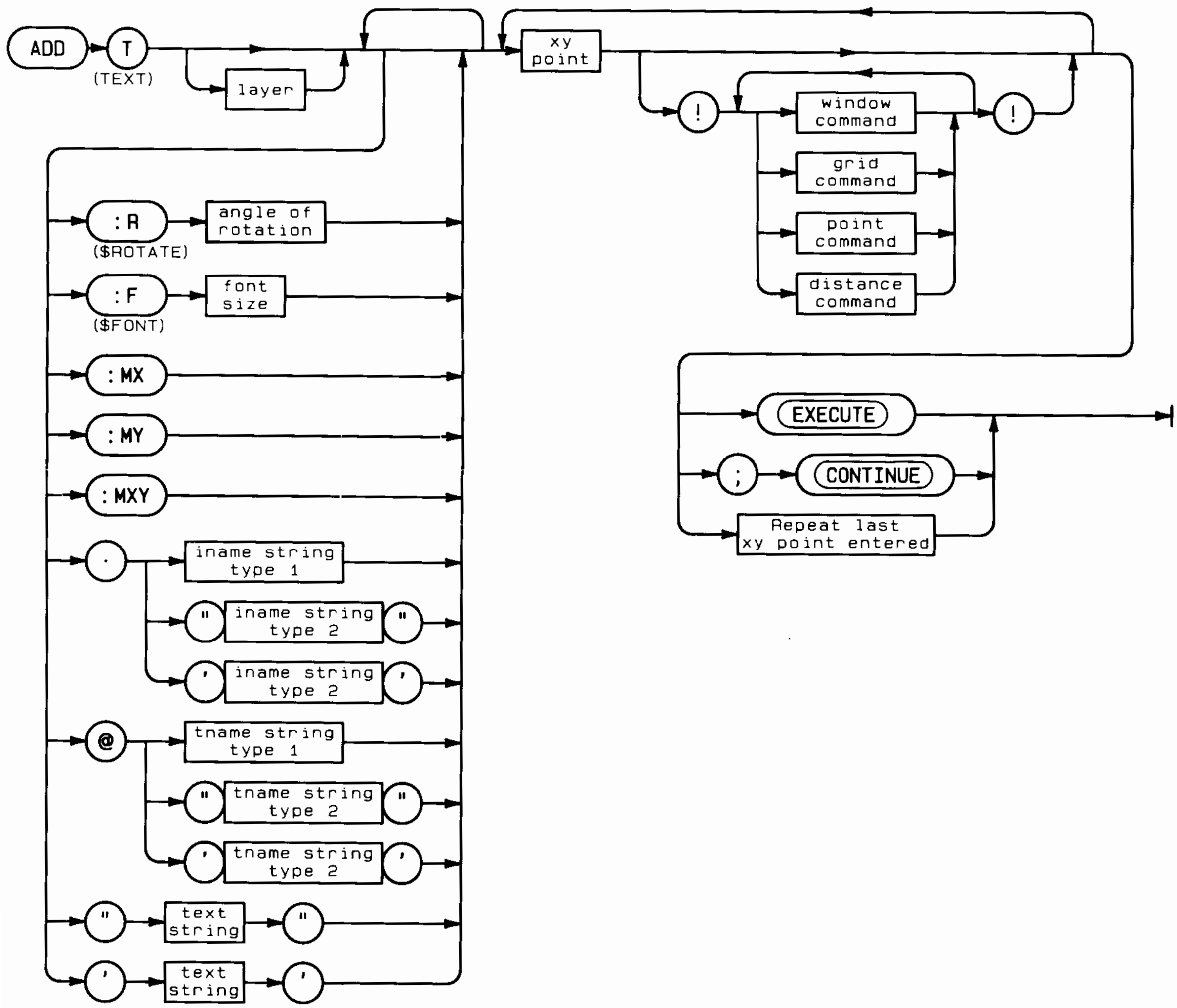
The rectangle below is added to the drawing. The first two locations caused the first representation of the rectangle to appear. Entering the third location caused the final representation of the rectangle.



ADD T (ADD TEXT)

Valid in:
Edit subsystem

The ADD command with a text descriptor allows a text component to be added to the drawing.



Note: the string must be specified at least once before attempting to enter an xy point

Item	Description/Default	Range Restrictions
layer	Layer number in which you wish to add the text. If the layer is not specified, the system defaults to the last layer number entered, and displays that number at the top of the screen.	1 - 255
angle of rotation	Angle, in degrees, at which you wish to rotate the text being added to the drawing. The default is 0 degrees.	−180.00 to 180.00
font size	Height of the characters , in current units, from which the text is to be formed. The default font size is 10 user units. The maximum font size is specified by the number of user units assigned to 16383 system grid points (a very large font size).	1 - (number of user units assigned to 16383 system grid points)
:MX	Specifies that the text is to be mirrored about the X axis, when added to the drawing.	
:MY	Specifies that the text is to be mirrored about the Y axis, when added to the drawing.	
:MXY	Specifies that the text is to be mirrored about both the X and the Y axes, when added to the drawing.	
iname string type 1	Set of characters forming the iname that you wish type 1 to assign to the text. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
iname string type 2	Set of characters forming the iname that you wish to assign to the text. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (%), (/), (\), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
tname string type 1	Set of characters forming the tname that you wish to assign to the text. All characters may be displayed and plotted. The first character must be either a letter (A-Z) or the underscore (_).	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
tname string type 2	Set of characters forming the iname that you wish to assign to the text. All characters may be displayed, but only those characters common to type 1 and type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (%), (/), (\), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
text string	Set of characters forming the text that you wish to add to the drawing.	a maximum of 64 characters may be entered. legal characters: A-Z, 0-9, (_), (*), (<), (>), ([), (]), (.), (,), (+), (−), (/), (\), (∧)

xy point	Location on the drawing at which you wish to add the text. The text is added such that the lower left corner of the text is placed at the xy point, and then is rotated about this point if rotation is specified. This point is called a trial placement location because entering a new xy point causes the text component to be moved from the previous location to the new location. The result is displayed on the CRT.	a maximum of 255 trial placement points may be entered.
!	Specifies to interrupt (suspend) the ADD command. Once the command is suspended, you may enter any of the following commands (complete with any of the legal options): the WINDOW command, the GRID command, the POINT command, or the DISTANCE command. This allows you to change the location of the window, change the current user or display grid, or obtain information (such as the xy value of a particular location via the POINT command). Re-entering ! allows you to continue entering the ADD command from the point at which it was interrupted.	

HELP syntax

```
ADD {desc[layer]|[I] devicename} [.iname][@tname] xy1 [xy2 ...] EOC
```

Use

The ADD command with a text descriptor allows a text component to be added to the drawing at the xy location specified. Entering additional xy points causes the text to shifted to the new location(s). Thus the location specified with xy point is a trial placement point.

The text is a "block letter" representation of the characters supplied. The text may be rotated about its origin, which is located at the lower left hand corner of the text. The size of the characters forming the text may also be specified. The text may be mirrored about the X and/or the Y axes.

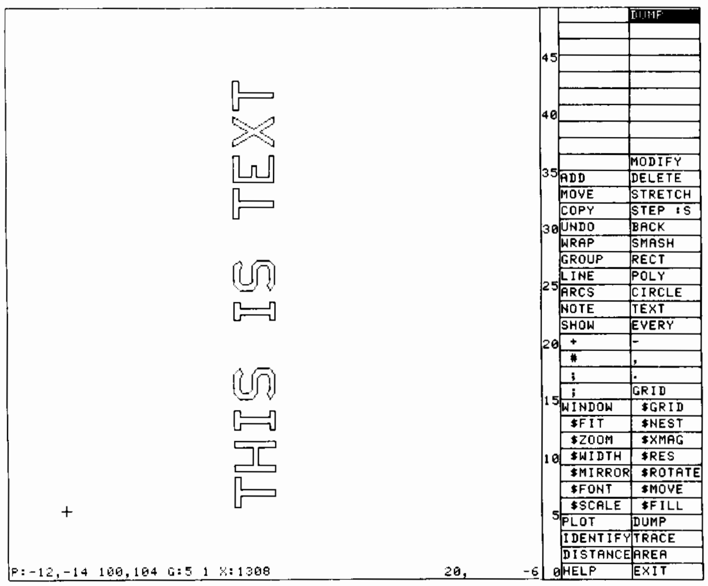
The interrupt option allows you to change the position of the window, change the current user or display grid, or obtain information. For example, if you find that the window is too small or is in the wrong position to add the text component, you may suspend the ADD command and then use the WINDOW command to reposition the window. Then, by entering ! again, you may continue to add the text component to your drawing by completing the command from the point at which it was suspended.

Example:

```
ADD T4  :R90  "THIS IS TEXT"  45,0 ;
```

result:

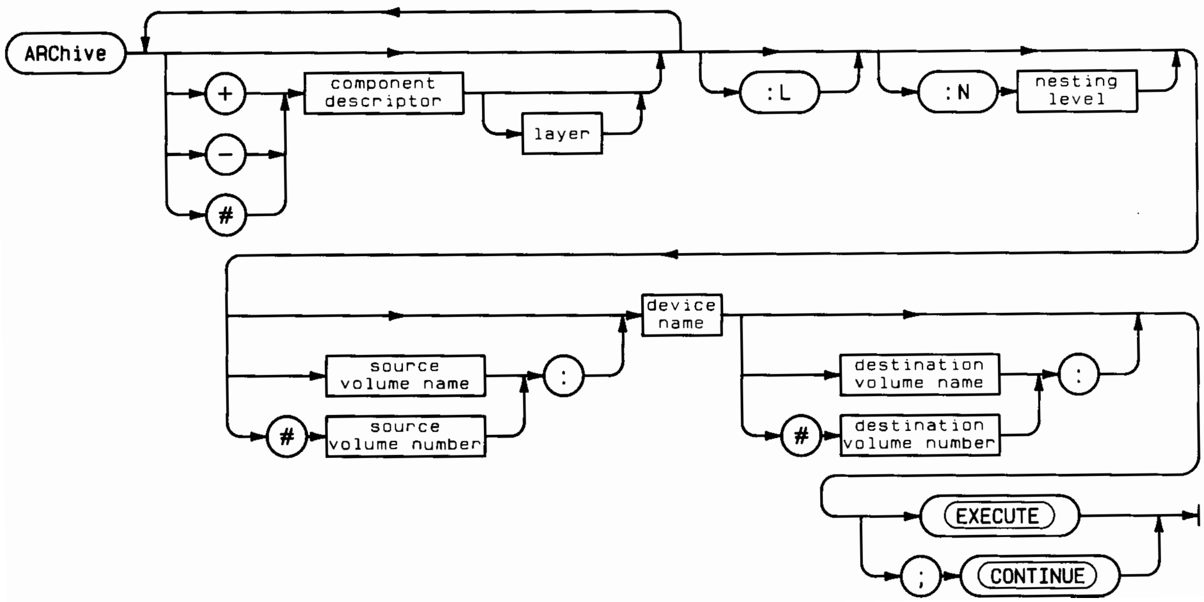
Text is added to layer 4 with characters 10 user units tall. The text is rotated 90 degrees about its origin and is placed at the location 45,0.



ARCHIVE

Valid in:
Graphics Editor

The ARCHIVE command provides a list of the commands of the Graphics Editor and its subsystems, used to create a drawing. This list may be output to the current system printer or stored in an EWE file, called an archive file. If the drawing contains instances, all instances in nesting levels 1 to the specified nesting level depth are included in the listing.



Item	Description/Default	Range Restrictions
+	Specifies to include in the archive file only those components specified by the descriptor on the designated layer. The default is all components on all defined layers.	
#	Specifies to include in the archive file only those components specified by the descriptor on the designated layer. The default is all components on all defined layers.	
–	Specifies to include in the archive file all components except those specified by the component descriptor and layer. The default is all components on all defined layers.	
component descriptor	One of the 9 defined descriptors.	A, C, E, I, L, N, P, R, T
layer	Layer containing the component(s) you wish to include or exclude in the archive file. If no layer is specified, the default is all layers.	0 - 255
:L	Specifies to output the archive file to the current system's printer as it is created.	
nesting level	Specifies to include components of devices displayed in nesting levels 1 to the specified depth. For example, :N3 specifies to include only those devices in nesting levels 1, 2, and 3 in the archive file created. The default is to include all nesting levels.	
Source volume name	Name of the volume containing the device file to be loaded.	
Source volume number	Number of the volume containing the device file to be loaded.	3 - 50
device name	The name of the device file which is to be archived.	
destination volume name	Name of the volume where you wish to store the archive file.	
destination volume number	Number of the volume where you wish to store the archive file.	3 - 50

HELP syntax

```
ARChive [(+|-|#)desc[layer]] [:L] [:Nlvl] [volume] name [volume] EOC
```

Use

Entering the ARCHIVE command along with the name of a drawing causes the system to create an EWE file called an archive file, that contains a list of the commands of the Graphics Editor and its subsystems, used to create the drawing. This file may be output to the current system's printer as well.

The hierarchy of the drawing is preserved. That is, the drawing is archived nesting level by nesting level, beginning with the deepest nesting level (deepest of the drawing or the deepest level specified with the :N parameter) and continuing to nesting level 1. All components devices displayed in the deepest nesting level are listed first, followed by all components of devices displayed in the next shallower nesting level, etc. This continues until all components of devices contained in nesting level 1 to the specified nesting depth are listed.

Using the +, −, # options allows you to limit the type of components and/or the layers of the drawing that are to be included in the archive file. The effects of these options are accumulative, and the intial state is to include all components on all layers. For example, to include only lines in layer 2 and arcs in all layers in the archive file, you would enter:

```
ARCHIVE -E +L2 +A ...
```

-E specifies to first exclude all components on all layers. +L2 specifies to now include lines on layer 2. +A specifies to include all arcs on all layers. Thus the accumulative effect of the limiting descriptors provides that only the desired components are included in the archive file.

Example

```
ARCHIVE :N2 EWSYS:TEST1 EWBOB:;
```

result:

a listing of the commands used to create the drawing named TEST1 is created. Components of devices in nesting level 2 are listed first followed by the components in nesting level 1.

The list is created from the drawing, TEST1, that is stored on the volume, EWSYS. The list is stored in an archive file, TEST1, on the volume, EWBOB.

Example

```
ARCHIVE EXAMP2 :L
```

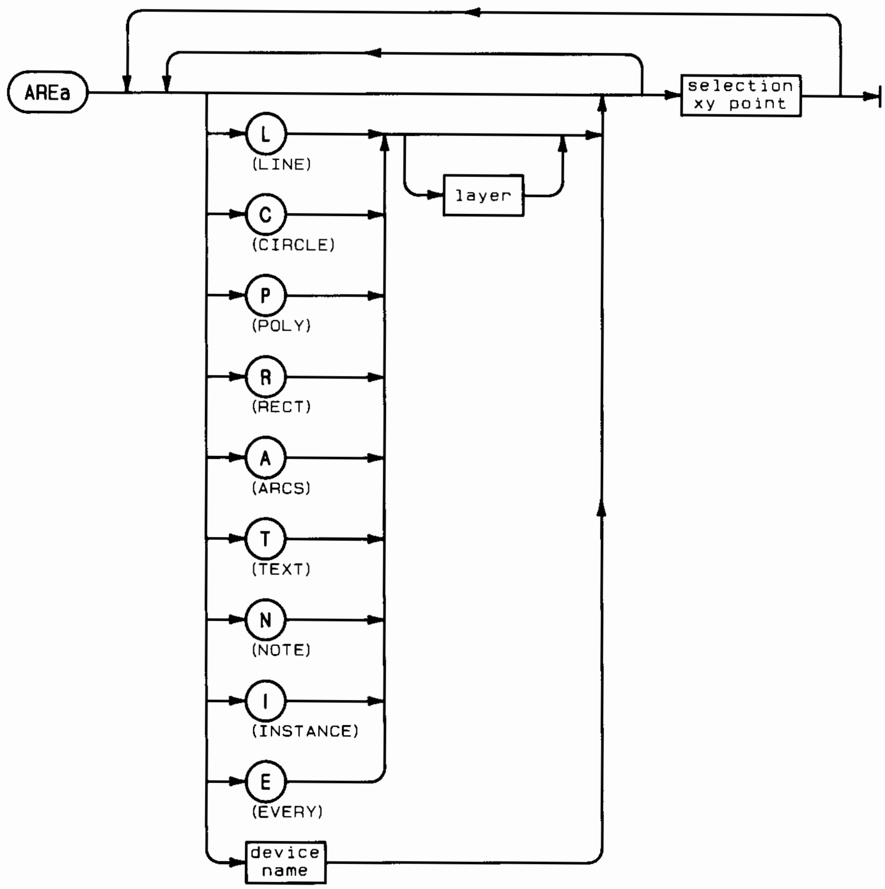
result:

a listing of the commands used to create the drawing named EXAMP2 is printed on the current system's printer.

AREA

Valid in:
Edit subsystem

This command calculates and displays the area occupied by the component selected. The total area of all components selected since the AREA command was invoked is also displayed.



Item	Description/Default
A, C, E, I, L N, P, R, T	Descriptor limiting the type of component that may be selected.
device name	Name of the instance whose area you wish to know. This limits selection to instance components with this device name.
layer	Layer containing the component whose area you wish to know.
selection xy point	Xy point lying in or on the component whose area you wish to know.

HELP syntax

```
AREa [{desc[layer]|devicename}] xysel1 [xysel2...]
```

Use

Enter the AREA command and any xy point lying inside or on the component whose area you wish to know.

When the AREA command is invoked two labeled values are displayed at the top of the CRT. These values are the area of the individual component selected and the accumulated area of all components selected since the AREA command was entered (labeled `AREA=` and `CUM=` , respectively). Re-entering the AREA command resets the accumulated area to zero. This information is erased when the screen is redrawn.

∧ (BACKUP)

Valid in:
Edit subsystem

The ∧ command may be entered to delete data points entered with any of the following commands (before the command is terminated): GROUP, WRAP, ADD, COPY, and WINDOW.

HELP syntax

none

Use

The ∧ command allows you to delete, or backup over, data points entered with a command. Data points are deleted in a last in first out fashion, hence the term backup. This command is valid only during the entry of another command. For example, entering `ADD L2 0,0 20,20 30,30 ^`, causes the point `30,30` to be deleted. The ADD command is still active and new data points may be entered (and backed over).

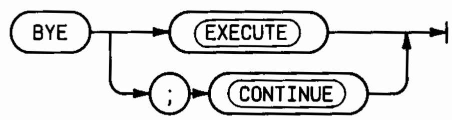
Backing over the first xy point entered with the command causes the system to display a message indicating an incomplete component has been entered. Though ominous in appearance, this message simply tells you that there are no more xy points to back over. You may continue entering the command by entering new xy points.

BYE

Valid in:
Graphics Editor
P.C. Board Layout

The BYE command allows you to exit the Graphics Editor, thus allowing access to the other modules. Any devices that were created in the edit subsystem that have not been saved are listed on the CRT. You may either return to the edit subsystem and save the individual devices or exit the Graphics Editor by re-entering the BYE command.

When BYE is entered in the PC Board Layout, a module creation of virtual plot files is terminated. The P.C. Board Layout menu is again displayed.



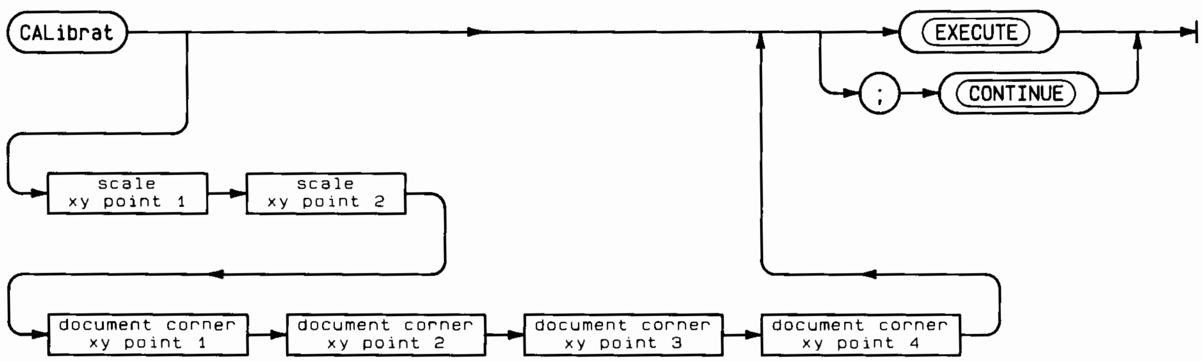
HELP syntax

```
BYE (terminates the EGS-45 Graphics Editor)
```

CALIBRAT

Valid in:
Edit subsystem

The CALIBRAT command allows a drawing to be digitzed from a graphic input device and to be scaled to the current drawing size. The command maps the current drawing area to the graphic input device platen.



Item	Description/Default
scale xy point 1 & xy point 2	This point along with scale xy point 2 specifies the scale in the X and Y direction of the drawing to be digitized from the graphic input device. For example, entering the points 0,0 and 100,80 sets the distance between the left edge of the drawing and the right edge of the drawing as 100 (100-0) current user units. Similarly, it sets the distance between the top edge and the bottom edge of the drawing as 80 (80-0) current user units. The location of the edges of the drawing are specified by the document corner xy points. If `CALIBRAT ;` is entered with no parameters, the values associated with the last complete CALIBRAT command are used as default locations and scale factors. The window is not adjusted as it is with a CALIBRAT command complete with parameters.
document corner xy point 1	Location of the lower left corner of the graphic input device active platen area. This point must be digitized from the graphic input device. The window is changed such that document corner xy point 1 "maps" to the lower left corner of the window.
document corner xy point 2	Location of the lower right corner of the graphic input device active platen area.
document corner xy point 3	Location of the upper left corner of the graphic input device active platen area.
document corner xy point 4	Location of the upper right corner of the graphic input device active platen area.

HELP syntax

```
CALibrat {[xy1 xy2 dig1 dig2 dig3 dig4] | EOC}
```

Use

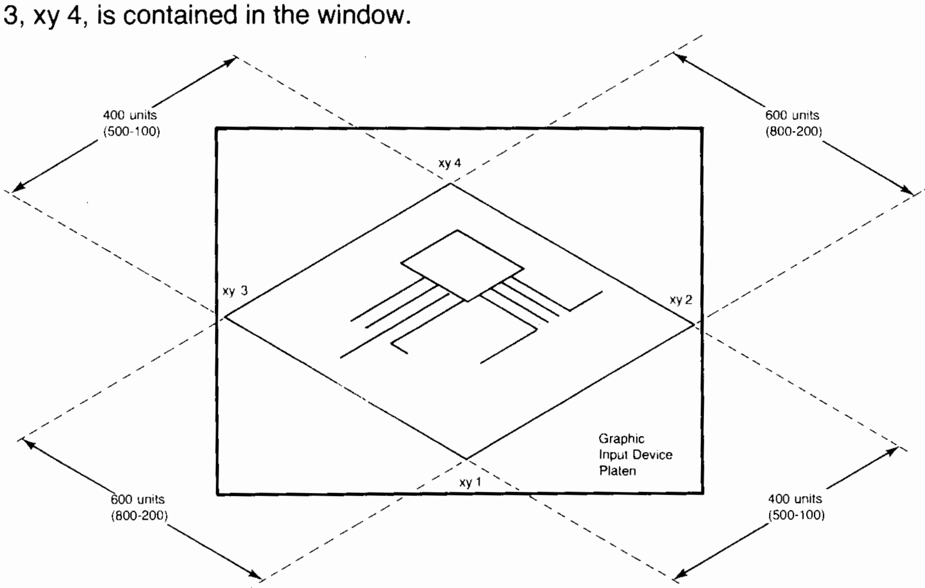
The CALIBRAT command allows a drawing to be digitized from a graphic input device and to be scaled to the current drawing size. The command maps the current drawing area to the graphic input device platen defined by the four xy points digitized with the command, until the calibrate mode is disabled. When in the calibrate mode, you are limited to accessing only the portion of the drawing area that maps to the graphic input device (the portion of the platen defined by the 4 xy locations). The system shows you when the calibrate mode is active by adding a character to the window location parameters in the lower left corner of the display. Normally this shows the window location in the following format: `P: x,y x,y` . In the calibrate mode this is changed to: `CP: x,y x,y`.

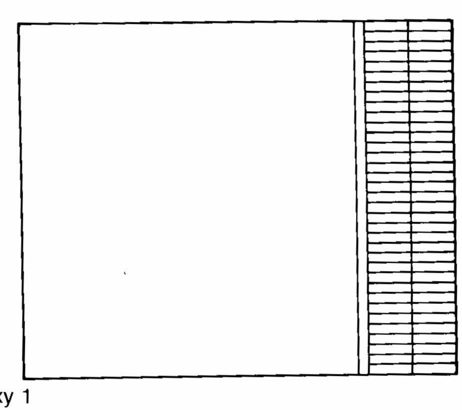
To disable the calibrate mode, enter any command which changes the current window parameters (such as `WINDOW :F` or `WINDOW :Z2`).

The drawing below illustrates the relationship of the 6 xy points that are entered with the CALIBRAT command.

```
CALIBRAT 100,200 500,800 xy 1 xy 2 xy 3 xy 4;
```

The window is adjusted such that the aspect ratio of the window is maintained. The lower left corner of the window maps to xy 1. The magnification of the drawing is changed so that the entire area defined by xy 1, xy 2, xy 3, xy 4, is contained in the window.

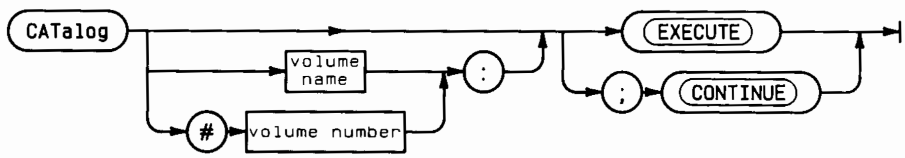




CATALOG

Valid in:
Graphics Editor

The CATALOG command will allows the user to obtain a printed list of the files contained in any EW volume currently on line. The list is output to the current system printer.



Item	Description/Default	Range Restrictions
volume name	Name of the volume you wish to catalog	6 characters max
volume number	Number of the volume you wish to catalog	3-50

HELP syntax:

```
CATalog [volume] EOC
```

Use

This command allows you to print the names of all files on a volume. Additionally, the file list shows the areas of the disc available for file storage. If no volume name is supplied, the current prefix volume is cataloged.

Example:

```
CAT EWSYS: ;
CAT #3: ;
```

Result and Explanation

```
File name          File size   Last update      Addr     LBS     File type
EWSYS:
SYSTEM.OS              40      15-Dec-81          10     512      Codefile
SYSTEM.FILER           31       7-Dec-81          50     512      Codefile
STREAM.STARTUP          1       9-Jul-82          81      68      Datafile
CEDIT.CODE             11      17-Dec-81          82     512      Codefile
MACRDATA.MAC            1      30-Sep-81          93     246      Datafile
SWSL_3A.DEV             1      28-Jun-82          94     270      Datafile
< UNUSED >              1                         95
SLIDERS.DEV             1      28-Jun-82          96     306      Datafile
GEDIT.CODE            298      21-May-82          97     512      Codefile
MENUDATA.MEN            2      21-May-81         395     468      Datafile
EXAMP.DEV               1      18-Jun-82         397     216      Datafile
EXAMP2.DEV              1      18-Jun-82         398     252      Datafile
DWG_3.DEV               5      23-Jun-82         399     274      Datafile
PROCDATA.PRO            3      25-Jun-82         404     120      Datafile
DIP.DEV                 1      28-Jun-82         407     396      Datafile
SLIDE1.DEV              1      28-Jun-82         408     198      Datafile
SLIDE2.DEV              1      28-Jun-82         409     252      Datafile
SWSL_2.DEV              1      28-Jun-82         410     234      Datafile
< UNUSED >              1                        411
SWSL_2A.DEV             1      28-Jun-82         412     288      Datafile
SWSL_3.DEV              1      28-Jun-82         413     234      Datafile
< UNUSED >              1                        414
19/19 files<listed/in-dir>, 402 blocks used, 3 unused, 1 in largest area
```

The sample catalog listing is shown above. It consists of:

- The name of a volume specifier identifying the volume being listed
- A list of all file names contained in the volume
- Each file's size in blocks (one block equals two records; one record equals 256 bytes)
- The last modification date
- The starting block address of the file within the volume
- The number of bytes in the last block of the file
- The file type (assigned by the system and determined by the file name suffix)

The first item listed in the catalog is the volume specifier identifying the volume being cataloged (EWSYS:). The first column contains the complete file name of the files stored on the volume (such as EXAMP.DEV and GEDIT.CODE). The second column contains a numeric value specifying the size of the file, in blocks (1 block = 512 bytes = 2 records). The third column identifies the date when the file was last modified or stored. The fourth column identifies the block number on the volume where the file begins (for example, EXAMP.DEV is a one block long file beginning at block number 381 on the system volume, EWSYS). The fifth column specifies the number of bytes used in the last block of the file (this specifies the number of bytes leftover in the 512 byte block). The last column identifies the type of file that is stored. Note that device, macro, process, and menu files are listed as Datafiles.

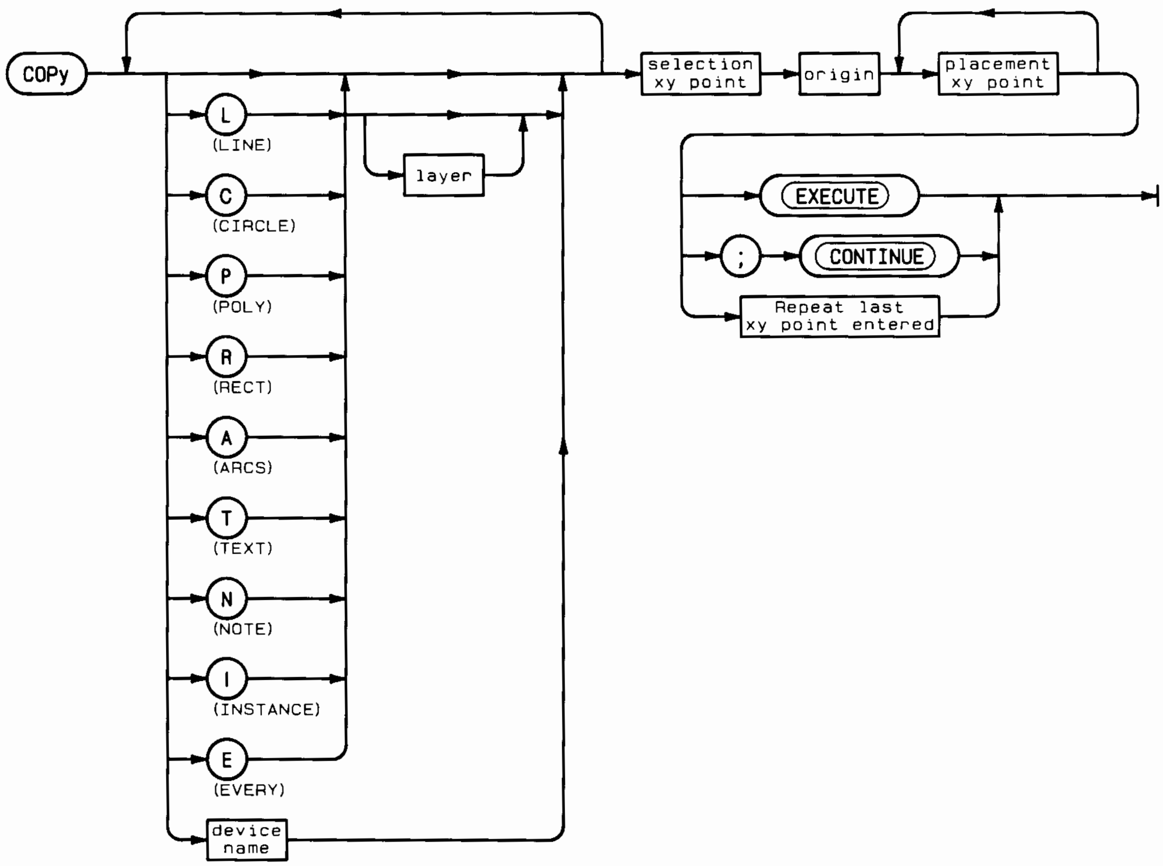
The last entry in the file name column is, <UNUSED>, which specifies that the area on the rest of the media has not been used. The number of blocks left on the media is listed along with the beginning block number of the unused area.

<UNUSED> also appears in the middle of the list of files stored on EWSYS. This indicates that a gap is present between the files. This is created when an existing file has been removed from the volume. Many such gaps may exist on a volume. By using the Pack function, you may pack all files on a volume together and eliminate this wasted space.

COPY

Valid in:
Edit subsystem

The COPY command allows you to make a copy of any component in the drawing area and place that copy at a new location. Multiple copies of a component may be made by simply specifying additional placement locations.



Item	Description/Default	Range Restrictions
A, C, E, I, L, N, P, R, T	Descriptor identifying the type of component to be selected and copied. The default descriptor is, E.	
layer	Layer containing the component to be copied.	1 - 255
device name	Name of a specific instance component to be copied.	
selection xy point	Xy location lying on or inside the component you wish to copy.	
origin	Xy location which acts as a reference point for copying. When a copy of the component is placed at the location specified, it is placed such that the origin exactly coincides with the placement location.	
placement xy point	Xy point specifying the location where the copy of the component is to be placed.	

HELP syntax

```
COPy [(desc[layer]|devicename)] xysel xyref xy1 [xy2...] EOC
```

Use

Entering the COPY command allows you to make a copy of a component by specifying an xy location lying on or inside the component. The optional descriptor and optional layer number limits selection of the component to be copied to the type specified by the descriptor. If the component to be copied is an instance component, you may limit selection by entering the name of the instance component to be copied. Limiting selection of the component to be copied is useful in congested areas of the drawing where selecting a specific component may be difficult. The descriptor (or device name) entered is valid for only one selection attempt.

Specifying the placement point causes the copy of the component to be added to the drawing such that its origin is placed exactly on the placement xy point. Entering additional placement xy points causes a copy of the component to be placed at each location.

Example:

```
COPY A 50,50 40,50 100,100 200,200 ;
```

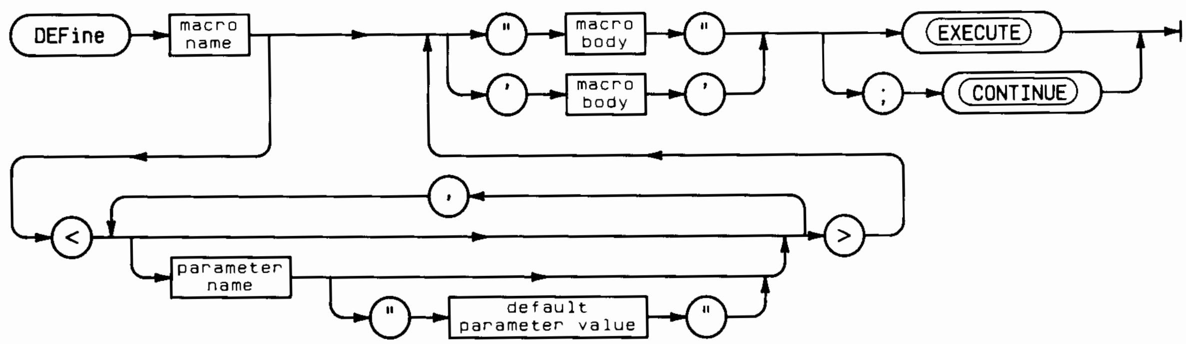
result:

An arc component (selection is limited to arc components by the descriptor A) located at 50,50 is selected to be copied and is assigned an origin at 40,50. Copies of this arc component are added to the drawing at the locations 100,100 and 200,200. These copies are added such that the origin of the copy coincides with the placement locations 100,100 and 200,200.

DEFINE

Valid in:
Macro subsystem

This command allows you to create new commands for use in the Graphics Editor (and all of its subsystems). The new command is composed of one or more existing: macros, Graphics Editor commands, edit subsystem commands, process subsystem commands, or macro subsystem commands.



Item	Description/Default	Range Restrictions
macro name	Name of the new command you are creating.	8 characters max. cannot be entire or portion of existing command name.
parameter name	Place holder for data to be entered with the newly created command.	12 characters max.
default parameter value	Default value for the parameter, if no data is entered with the newly created command.	8 characters max.
macro body	A string of commands and parameters which form the macro.	

HELP syntax

```
DEFine name [<pram1 ["dflt1"] ...>] "body"
```

Use

The DEFINE command allows you to create a new command from existing commands of the Graphics Editor, any of its subsystems, or from existing macros. Enter the DEFINE command along with the name of the new macro.

You cannot enter a macro name that could be interpreted by the system to be a command name. For example, entering `DEFINE WIND` ... could cause the system to reject the DEFINE command since WIND is a valid entry for the WINDOW command.

If the macro name entered is already defined, the new macro definition replaces the previous macro definition.

The parameter name(s) entered with the DEFINE command create a space holder in the new command for required or optional data (used by the commands comprising the new command). Each of these parameters may be assigned a default value.

For example, to create a new command that adds a 10 unit wide line to the drawing, the DEFINE command might look like:

```
DEFINE WID_LINE <LAYER"1"> "ADD L<LAYER> :W10";
```

This creates a macro named WID_LINE. Entering

```
WID_LIN 2  0,0  100,100
```

is equivalent to entering

```
ADD L2 :W10  0,0  100,100
```

as both cause a 10 unit wide line to be added to layer 2 between the points 0,0 and 100,100. The parameter layer is assigned the first data value entered with the macro WID_LINE and passes this value into the macro body. Thus entering `WID_LIN 2` ... assigns the value 2 to the parameter, `layer`. This value is then passed into the macro body to the corresponding parameter, `layer`, thus causing the line to be added to layer 2.

If a macro is created to replace a descriptor, it is necessary to insert a blank before the descriptor in the macro body (see the example below).

Macros may be defined using other macros; however, they must be defined in the same macro file (since only one macro file may be loaded at a time, and since the macro file in which it is stored must be loaded into the system before the macro may be used).

Example:

```
DEFINE LINE  " L"
```

This creates a macro named LINE that may be used to replace the line descriptor L. Notice that a space is included before the line descriptor in the macro body.

Example:

```
DEFINE FIT  "WINDOW :F ;"
```

This creates a macro named FIT that may be used to adjust the window such that the drawing on the CRT just exactly fits into the window area.

Example:

```
DEFINE STRTUP "MACRO MINE ; EXIT ; PROCESS MYPROC ; EXIT ;"
```

This creates a macro named STRTUP that may be entered from the Graphics Editor to load the macro file named MINE and the process file named MYPROC. Attempting to enter this macro from any of the subsystems would cause an error to be generated since the commands that form the body of the macro are valid only in the Graphics Editor.

Example:

```
DEFINE EXAMPLE <A,B,C>  "ADD T<A> :F<B> :R<C> 'TEXT' 0,0;
NEWEXAMP <A>,<B>"

DEFINE NEWEXAMP <X,Y> "ADD N<X> :F<Y> 'NOTE' 0,10;"
```

This shows how one macro can use another macro in its definition. It also shows how values entered with one macro are passed to another macro. Entering: `EXAMPLE 1, 20, 90` causes the text component `TEXT` to be added to layer 1 of your drawing. The text is 20 user units tall and is rotated at an angle of 90 degrees. The three values entered with the macro are passed into the macro body as A, B, and C, respectively.

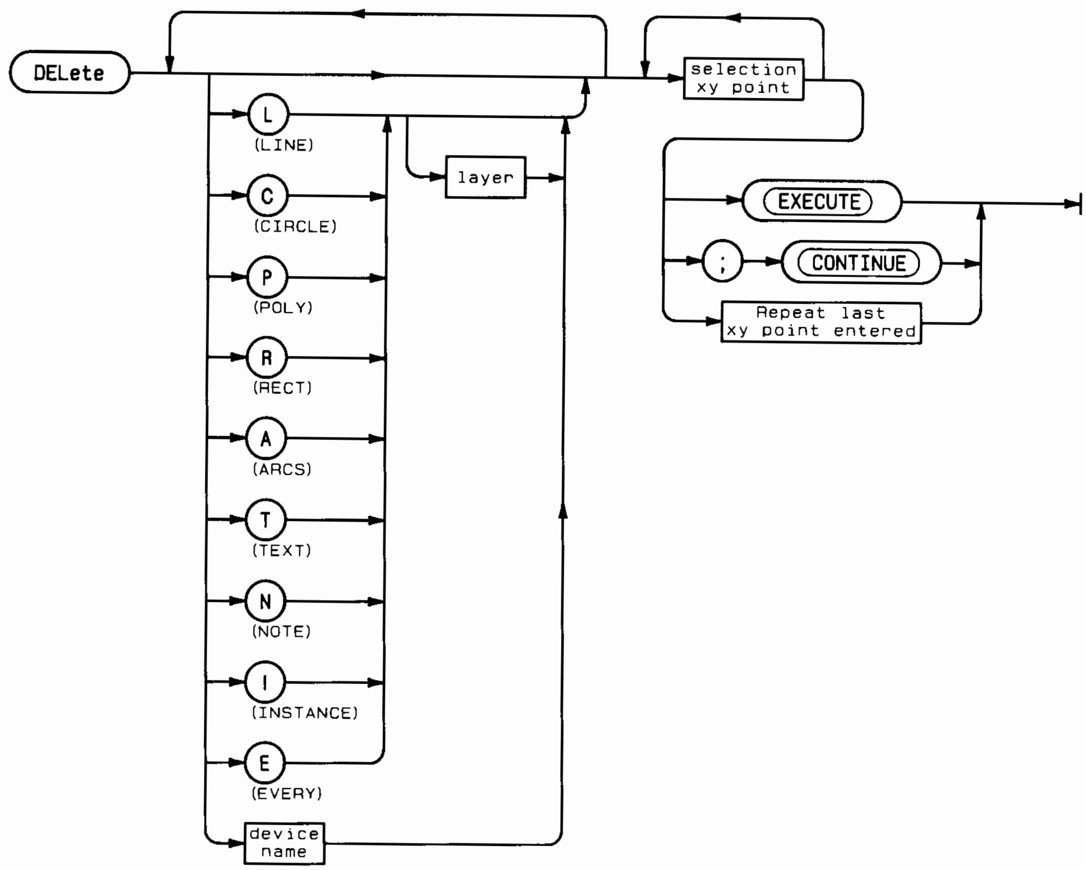
After the text is added to your drawing, the system executes the next command in the macro body, which is the macro NEWEXAMP. The values assigned to A and B are passed to the macro NEWEXAMP and are assigned as the values of X and Y in NEWEXAMP. This causes the note component `NOTE` to be added to layer 1 of your drawing with a font size of 20 (the values passed to NEWEXAMP by EXAMPLE). The note is added at the xy location 0,10.

For further examples, list the contents of the supplied macro file, MACRDATA.

DELETE

Valid in:
Edit subsystem

The DELETE command allows any component to be removed from the drawing.



Item	Description/Default	Range Restrictions
A, C, E, I, L, N, P, R, T,	Limits selection of the component to be deleted to the component type specified (such as R for rectangle).	
layer	Layer containing the component to be deleted.	1 - 255
device name	Device name of the instance that you wish to delete.	
selection xy point	Point lying in or on the component to be deleted. Entering additional selection points allows you to re-select the component to be deleted. No component is deleted until the command is terminated.	

HELP syntax

```
DELete [(desc[layer]|devicename)] xysel1 [xysel2...] EOC
```

Use

The component to be deleted is selected by specifying a point in or on the component. The layer and descriptor (or device name for instance components) may be used to limit selection of the component to be deleted. Entering additional selection points allows you to re-select the component to be deleted.

Example:

```
DELETE L2 100,100 ;
```

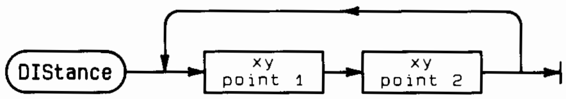
result:

A line in layer two passing through the point 100,100 is deleted from the current drawing.

DISTANCE

Valid in:
Edit subsystem

The DISTANCE command allows you to display the distance between two specified points, in user units.



HELP syntax

```
DIStance xy1 xy2
```

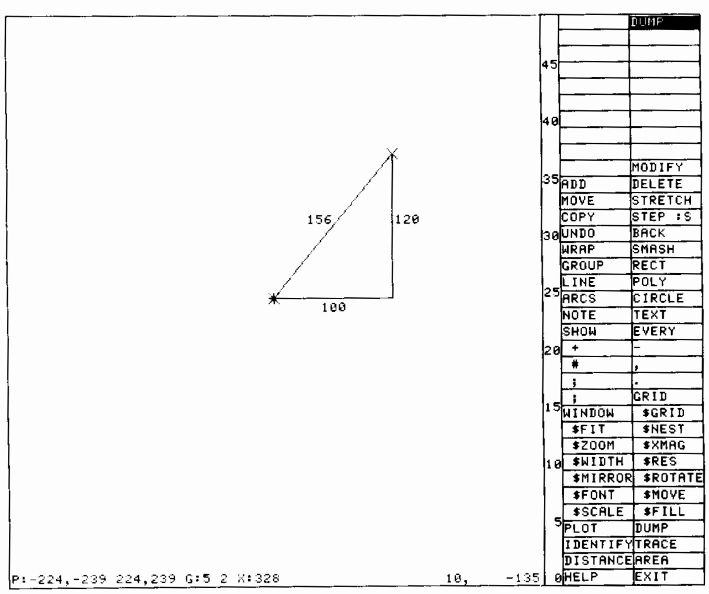
Use

Enter the DISTANCE command and select two xy locatons. The x distance, y distance, and the vector distance between the two points is then displayed. This information is erased when the screen is redrawn.

Example:

```
DISTANCE 0,0  100,120
```

result:



DUMP

Valid in:
Edit subsystem

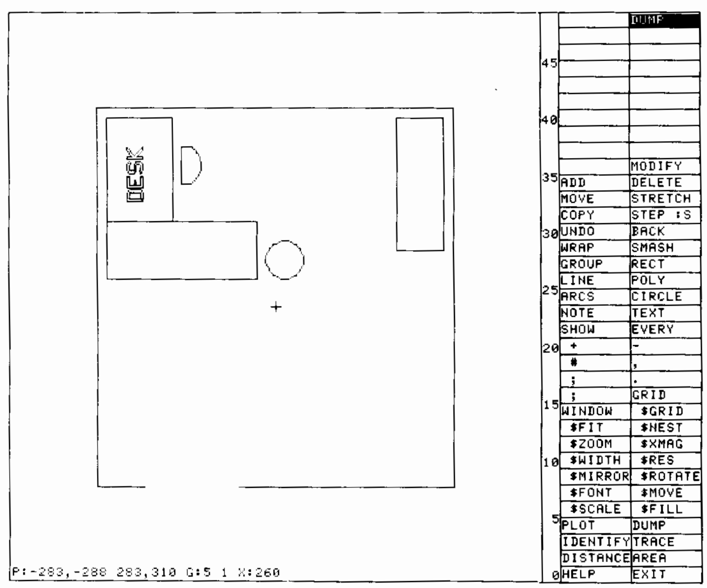
The DUMP command allows you to obtain a printed copy of the entire graphics screen. The copy is produced on the computer's built-in thermal printer.

HELP syntax

DUMp (graphics screen dump to internal printer)

Example:
DUMP

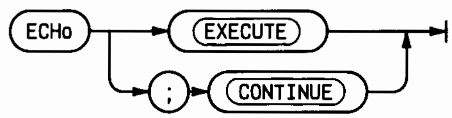
result:



ECHO

Valid in:
Graphics Editor
Edit subsystem
Macro subsystem
Process subsystem

The ECHO command allows you to enable/disable echo. When echo is enabled, each command entered is printed on the current system printer.



HELP syntax:

```
ECHo (toggles on/off; On writes all input to printer)
```

Use

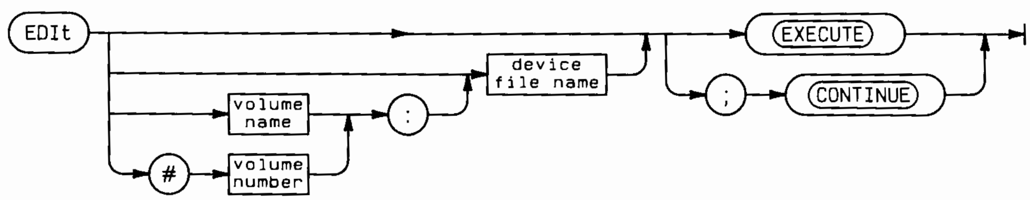
If echo is disabled, entering ECHO enables it. If echo is enabled, entering ECHO disables it.

When echo is enabled, it is enabled in the Graphics Editor and all of its subsystems.

EDIT

Valid in:
Graphics Editor

The EDIT command allows you to load a particular device file (file containing a drawing that you wish to modify) and provides access to the edit subsystem. Once in the edit subsystem, you may use any of the commands valid there to modify an existing drawing, to plot a drawing, or to create a new drawing.



Item	Description/Default	Range Restrictions
volume name	Name of the volume where you wish to get the device file.	
volume number	Number of the volume where you wish to get the device file	3-50
device file name	Name of the file in which you want to edit. The first character of the file name must be a letter (A-Z) or the underscore character (_).	legal characters: A-Z, 0-9, (_)

HELP syntax:

```
EDIt [volume] <filename | EOC>
```

Use

Entering the EDIT command with and without a device name can cause varied results.

- If no drawing (device name) is specified with the EDIT command, a blank drawing is loaded as you enter the edit subsystem. This allows you to create a new drawing.
- If a file name is supplied, the result depends upon whether a source volume specifier is entered and whether a device file with the supplied name exists.

  If a source volume specifier is supplied, the system searches only the volume specified for the device file. If no source volume specifier is supplied, the system searches all volumes currently on line, beginning with the system volume, EWSYS, until a device file with the supplied name is found.

  If a drawing is specified with the EDIT command and a drawing with that name is found, the device file containing the drawing is loaded as you enter the edit subsystem. The drawing is displayed on the CRT and is ready for modification through the commands of the edit subsystem.

  If the device file is not found, the system loads a blank drawing as you enter the edit subsystem. No device file is loaded; however, the file name entered then becomes the current drawing (device file) name.

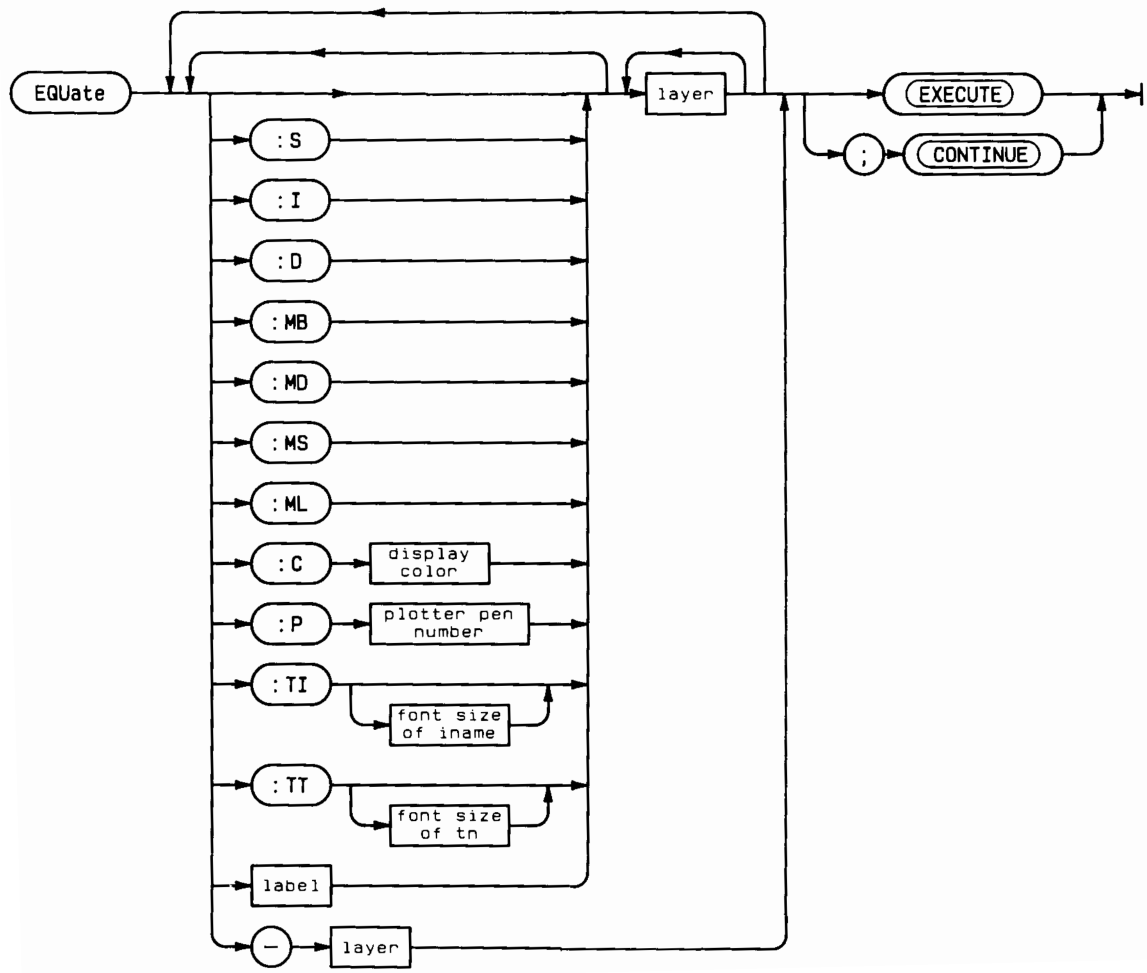
Example:

```
EDIT ;
EDIT #3:EXAMPL
EDIT EWSYS:EXAMPL
```

EQUATE

Valid in:
Process subsystem

This command allows you to define each layer that comprises the drawing area in the edit subsystem. **Layer 0** is reserved by the system for displaying instances of library parts and **must always be defined.**



Item	Description/Default	Range Restrictions
S	Specifies that the layer is a symbolic layer.	
:I	Specifies that the layer is an interconnect layer.	
:D	Specifies that the layer is a detail layer. The default specifies that the layer is a detail layer.	
:MB	Specifies that components entered on this layer are drawn with a broken line type.	
:MD	Specifies that components entered on this layer are drawn with a dotted line type.	
:MS	Specifies that components entered on this layer are drawn with a solid line type.	
:ML	Specifies that components entered on this layer are drawn with a long dashed line type. The default is :MS.	
display color	Single character specifying the color that components entered in this layer are to be drawn with on the CRT (this is applicable only for systems with color displays). The default color is green.	A = aqua B = blue G = green P = purple R = red Y = yellow W = white
plotter pen	Plotter pen number with which to plot components in this layer. The default pen number is 0.	0 - 8
:TI	Specifies to plot all inames of components entered in this layer, on the CRT. In this fashion, the iname is always visible. The default is to have this option disabled.	
font size	Specifies the height of the characters with which the inames of components entered on this layer are to be plotted on the CRT. The font size represents the size in user units, as defined with the UNITS command in the process subsystem. The default font size is 1 user unit.	maximum is the number of user units assigned to 16383 system grid points.
:TT	Specifies to plot all tnames of components entered in this layer, on the CRT. In this fashion, the tname is always visible. The default is to have this option disabled.	
font size	Specifies the height of the characters with which the inames of components entered on this layer are to be plotted on the CRT. The font size represents the size in user units, as defined with the UNITS command in the process subsystem. The default font size is 1 user unit.	maximum is the number of user units assigned to 16383 system grid points.
label	8 character label which may be used as a note to describe the contents of the layer.	
layer	Number of the layer being defined. Entering a negative layer number causes the current definition of that layer to be deleted.	−255 to 255

HELP syntax

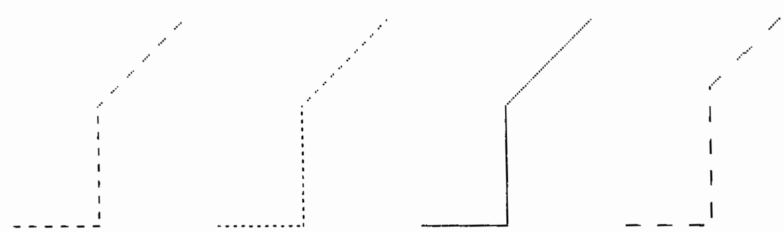
```
EQUate [:<S|D|I>] [:Ccolor] [:Ppen] [:M<S|D|B|L>] [:T<I|T>[font]]
[label] layer
```

Use

The EQUATE command allows you to define one or more layers of the current process file. Unless the process file is saved before exiting the Graphics Editor, the new layer definitions are lost.

The type of layer (symbolic, detail, or interconnect) corresponds to the display options with the WINDOW command. For example, `WINDOW :I` specifies to display only those components entered on interconnect layers (those layers defined with `EQUATE :I`).

Components may be drawn with one of four line types:



:MB	:MD	:MS	:ML
(broken)	(dotted)	(solid)	(long dash)

The pen number identifies which pen the HP plotter will use when plotting components entered on this layer. The HP 7225 plotter ignores the pen number assigned to a layer since it is a single pen plotter.

The :TI and :TT options cause the inames and tnames of components entered on the layer to be plotted on the CRT, whenever the layer is turned on. While a large number of different characters may be used for the iname or tname assigned to a component (and displayed with the IDENTIFY and TRACE commands) only the following characters are plotted on the CRT:

A-Z, 0-9, (_), (<), (>), (/), (\), (∧), (*), (+), (,), (−), (.), ([), (]).

All other characters are plotted on the CRT as a small box.

Example:

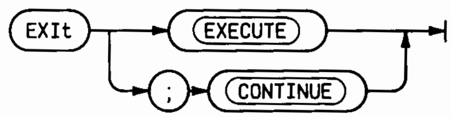
```
EQUATE :D :MB :CY :P3 NPUT 1,2,3
```

This defines layers 1, 2, and 3 to be detail layers. Components entered on these layers are drawn in yellow (on systems with a color CRT) and are drawn with a broken line type. When plotted, components on these layers are plotted using pen number 3. The label NPUT is assigned to each layer.

For further examples, list the contents of the current process file by using the LIST command.

EXIT

Valid in:
Graphics Editor
Edit subsystem
Macro subsystem
Process subsystem



HELP syntax

```
EXIt
```

Use

The EXIT command allows you to exit the subsystem you are currently in. The Graphics Editor prompt, MAIN> indicates that you have successfully exited the subsystem and have returned to the Graphics Editor.

If the file you are working on the subsystem has been modified but not saved, the system issues a warning to that effect. You may then either save the file, modify the file further, or re-enter the EXIT command to exit the subsystem without saving the file.

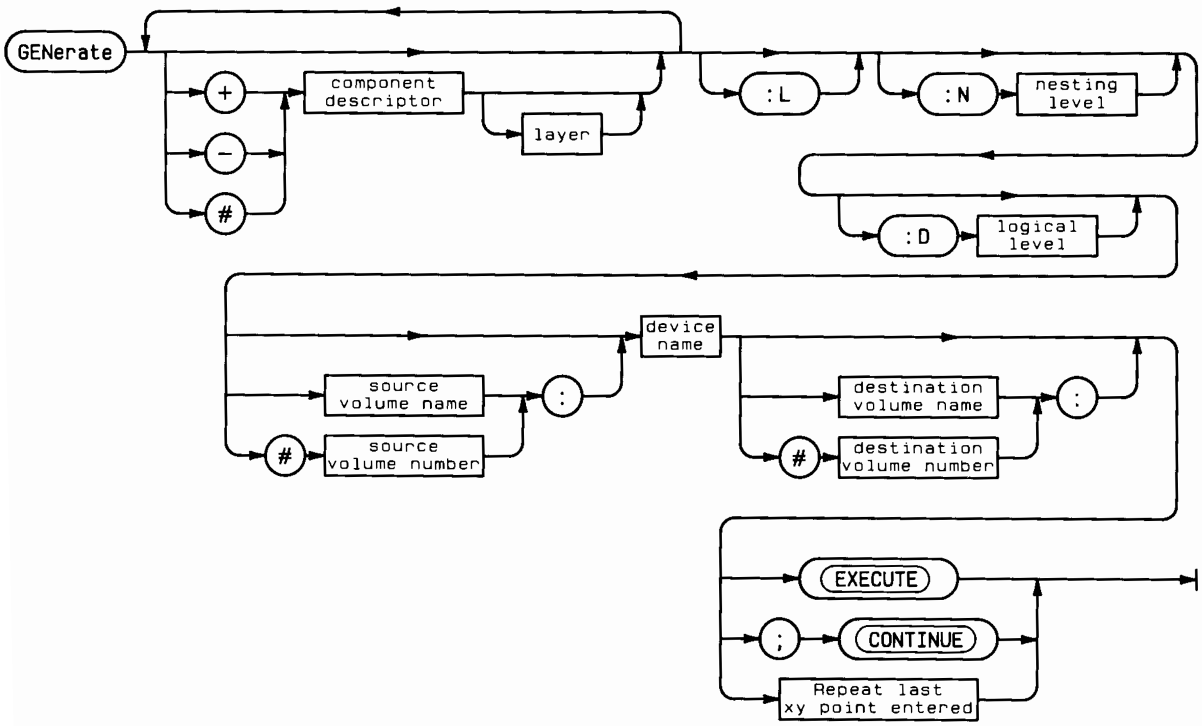
Example:

```
EXIT ;
```

GENERATE

Valid in:
Graphics Editor

The GENERATE command provides a list of the commands of the Graphics Editor and its subsystems, used to create a drawing. This list may be output to the current system printer and stored in an EWE file, called a generate file. If the drawing contains instances, all instance components are smashed (i.e. the link to a device is replaced with the components that form the device) and the components comprising the devices are included in the generate file. By specifying a maximum nesting depth you can limit the devices that are included in the generate file. Devices in deeper nesting levels are not included in the generate file.



Item	Description/Default	Range Restrictions
+	Specifies to include in the generate file only those components specified by the descriptor on the designated layer. The default is all components on all defined layers.	
#	Specifies to include in the generate file only those components specified by the descriptor on the designated layer. The default is all components on all defined layers.	
–	Specifies to include in the generate file all components except those specified by the component descriptor and layer. The default is all components on all defined layers.	
component descriptor	One of the nine defined descriptors.	A, C, E, I, L, N, P, R, T
layer	Layer containing the component(s) you wish to include or exclude in the generate file. If no layer is specified, the default is all layers.	0-255
:L	Specifies to output the generate file to the current system's printer as it is created.	
logic level	Specifies to output the generate file such that devices which are less than or equal to the logical level specified are not smashed. The default is –32 676. See level command for explanation.	
nesting level	Specifies to include components of devices displayed in nesting levels 1 to the specified depth. For example, :N3 specifies to include only devices displayed in nesting levels 1, 2, and 3 in the archive file created. The default is to include all nesting levels.	
Source volume name	Name of the volume containing the device file to be loaded.	
Source volume number	Number of the volume containing the device file to be loaded.	3-50
device name	The name of the device file which is to be generated.	
destination volume name	Name of the volume where you wish to store the generate file.	
destination volume number	Number of the volume where you wish to store the generate file.	3-50

HELP syntax

```
GENerate [(+|#|-)desc[layer]] [:L] [:Nlvl] [:D log_lvl][volume] name
[volume] (name|EOC)
```

Use

Entering the GENERATE command along with the name of a drawing causes the system to create a special file called an generate file, that contains a list of the commands of the Graphics Editor and its subsystems, used to create the drawing. This file may be output to the current system printer as well.

Using the +, −, # options allows you to limit the type of components and/or the layers of the drawing that are to be included in the generate file. The effects of these options are accumulative, and the intial state is to include all components on all layers. For example, to cause only components in layer 1 to be included in the generate file you would specify −E +E1. −E specifies to exclude all components in all layers. +E1 specifies to include all components in layer 1 in the generate file.

Example:

```
GENERATE :L DRAW1 EW5:
```

result:

A generate file containing a list of the commands used to create the drawing, DRAW1 is created. All instances contained in the drawing are smashed (i.e. replaced with the components forming the device they reference) and their constituant components included in the generate file.

A copy of the list of commands forming the generate file is printed on the current system printer as the file is created. The system assumes that the drawing, DRAW1, is stored on the system volume, EWSYS, since no source volume specifier is supplied. However, if the drawing is not found there, the system will check all volumes on line until the device file is found or until it determines that no such file exits.

The generate file that is created is named DRAW1 and is stored on the volume, EW5.

Example:

```
GENERATE -E +E1 +L2 :N3 EW2: DRAW1 EW3:
```

result:

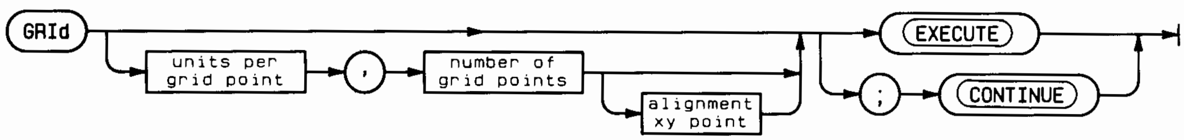
A generate file containing a list of the commands used to create the drawing, DRAW1 is created. The drawing is stored on the volume EW2. Only components of devices displayed in nesting levels 1, 2, and 3 are considered when creating the generate file. All components contained in layer 1 and line components contained in layer 2 of the drawing are included in the generate file, all other components are ignored.

The generate file that is created is named DRAW1 and is stored on the volume, EW3.

GRID

Valid in:
Edit subsystem

The GRID command allows you to determine the coarseness of the user grid, the alignment of the grid, and the portion of the grid that is displayed (the display grid). All xy points entered are "snapped" to the nearest user grid point, whether displayed or not.



Item	Description/Default
units per grid point	Number of user units between consecutive user grid points. User units are specified in the process file.
number of grid points	Determines the number of user displayed. grid points If the number 5 is entered, then every 5th user grid point is displayed. If the number 8 is entered, then every 8th user grid point is displayed. This is not to be confused with the user grid. Xy locations entered are snapped to the user grid, not the display grid.
alignment point	Shifts the grid such that a grid point is located exactly on the alignment point. The default alignment point is the drawing origin 0,0.

HELP syntax

```
GRId [gridspace,gridmult [xypnt]] EOC
```

Use

Entering only the GRID command with no options toggles the display grid on or off. If the grid was on, entering GRID ; turns the display of the grid off. If the grid was off, entering GRID ; turns the display of the grid on.

The user grid setting may affect entry of new grid parameters. For example, attempting to align a new grid with a point that is not on the user grid causes the alignment point to "snap" or shift to the nearest user grid point, thus keeping the desired alignment from occurring. It may be necessary to change the user grid spacing to allow alignment with the xy location that you want.

Example:

```
GRID  10,2 ;
```

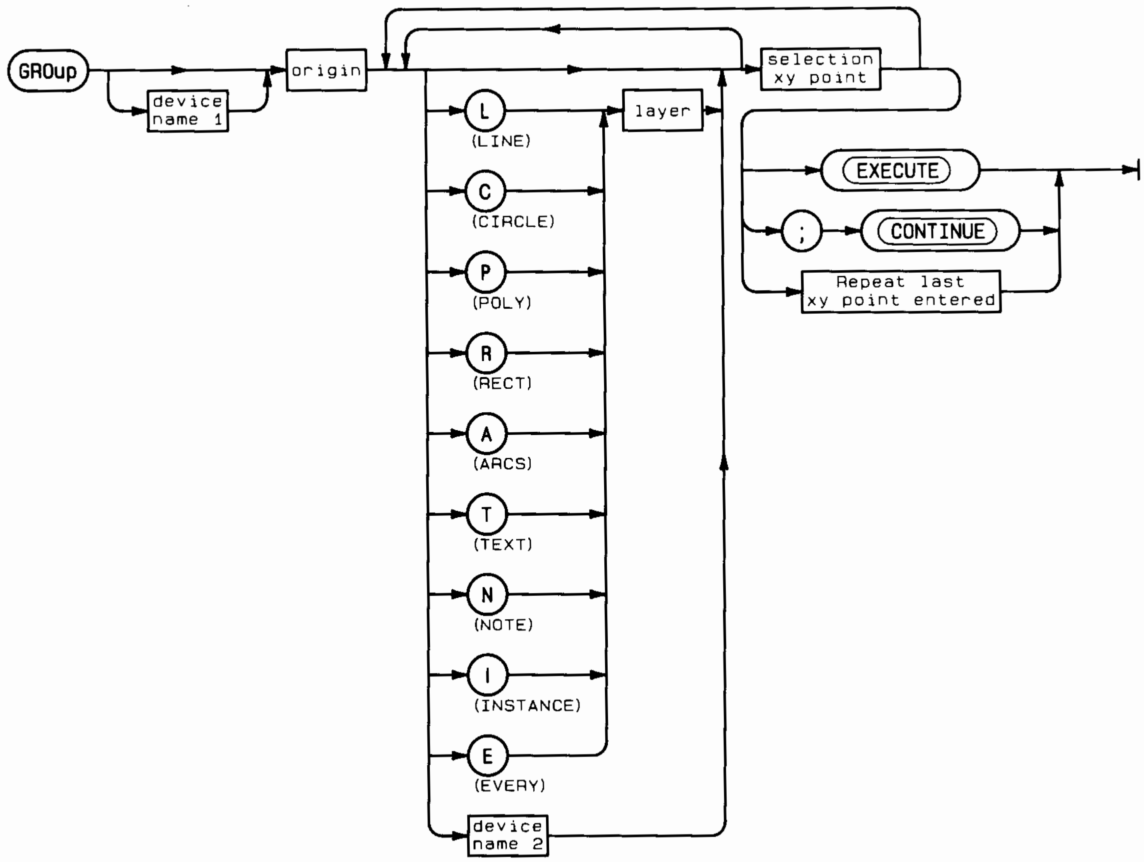
result:

A new user grid is set such that 10 user units exist between each user grid point and every 2nd user grid point is displayed. If the current user units are inches, then distance between two user grid points is 10 inches.

GROUP

Valid in:
Edit subsystem

The GROUP command allows you to create a new device from selected components displayed in the drawing area. A copy of the new device is placed in computer memory. The components forming the new device are displayed in nesting level 2. An instance component replaces the components in nesting level 1, linking the new device to your drawing.



Item	Description/Default	Range Restrictions
device name 1	Name assigned to the new device.	8 characters max. legal characters: A-Z, 0-9, (_)
origin	Xy location specifying the origin of the device that you create.	
A, C, E, I, L, N, P, R, T	Descriptor identifying the type of component to be selected. The default descriptor is E.	
layer	Layer containing the component to be selected.	1 - 255
device name 2	Name of the instance component that you wish to select. This limits component selection to instance components with the supplied name.	
Selection xy point	Xy location lying inside or on the component you wish to select. Entering additional selections allows additional components to be included in the device that you create.	

HELP syntax

```
GROup [devicename] xyorg [(desc[layer]|devicename)] xysel ... EOC
```

Use

The GROUP command allows you to create a device that consists of individual components selected from the drawing area of the display. Unlike the WRAP command, where the device is composed of all components in a specified area (the wrapping rectangle), the GROUP command allows selection of components from any location on the screen.

Specifying a descriptor (and optionally, a layer) before selecting a component limits the type of component that can be selected. If the component to be selected is an instance component, you may limit component selection by entering the name of the linked device. This allows only instance components with the supplied name to be selected. This is useful in areas of the drawing where components are dense and it is difficult to accurately select the component. The descriptor entered is valid for only one selection.

Entering a name with the GROUP command causes a library part to be created from the device when the drawing is saved. Additionally, assigning a name to the device allows an instance of the device to be added to other sections of the drawing with the ADD command. If the name entered already exists as a device name, the system will not allow you to use that name and the command must be re-entered.

Additionally, you cannot enter a name for the device that could be interpreted by the system as a command name or a macro name. For example, entering `GROUP WIND` ... would cause the system to reject the GROUP command since WIND is a valid entry for the WINDOW command. Similarly, if a macro named ARCS is defined, entering `GROUP ARCS` ... would cause the system to reject the WRAP command.

If no device name is entered, the device created cannot be stored and cannot be added to the drawing (no library part is created when the drawing is saved with the SAVE command). However, an instance of the device can still be copied, moved, smashed, etc., just like any other component. The device is present in computer memory until you exit the Graphics Editor, and then is lost. When the drawing is saved, the instance of the device is smashed, thus the components forming the linked device are placed back in nesting level 1 with your drawing.

The origin specified with the command is used to accurately place an instance of the device. For example, after a device is created, adding an instance of the device to your drawing at the location 10,10 causes the instance component to be added such that its origin is placed exactly at the location 10,10.

Example:

```
GROUP FUN  0,0  L 10,10  C2 30,10  50,20  ;
```

result:

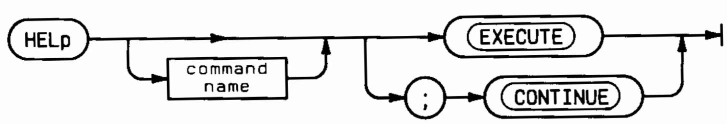
A device named FUN is created with its origin located at 0,0. The components that form the new device consist of: a line which passes through the location 10,10 (only a line component could be selected since the descriptor L is specified before the xy point is entered), a circle on layer 2 which contains the point 30,10 (again only a circle component on layer 2 may be selected at this location because of the descriptor and layer), and the device located at the point 50,20.

The components that form the device FUN are replaced in your drawing with an instance component that links your drawing to the new device. The components forming FUN are displayed in nesting level 2.

HELP

Valid in:
Graphics Editor
Edit subsystem
Macro subsystem
Process subsystem
P.C. Board Layout

The HELP command provides syntax information about a single command, a macro, or all commands available in the Graphics Editor or one of its subsystems. When HELP is entered in the P.C. Board Layout module, help syntax for all valid commands is printed on the built-in printer.



Item	Description/Default	Range Restrictions
command name	Name of the command or macro for which you seek help. The default is all valid commands.	Must be valid command in the part of the system from which it is entered

HELP syntax

```
HELp <commandname|macroname|EOC>
```

Use

Entering the HELP command along with the name of a valid command causes the system to display a help message detailing the syntax of that command. The help message for each command is provided in the command reference along with the syntax diagram for that command so that you may compare the format of the help message with that of the pictorial representation.

You may obtain help only for commands valid in the Graphics Editor subsystem from which the HELP command is entered.

The syntax appearing in the help messages is described below.

`[ ]`	Items enclosed in square brackets are short notes describing optional parameters or data.
`{ }`	items enclosed in braces represent a choice of items, where at least one of the items must be entered.
`¦`	Separates choices of parameters, data, or items to be entered with a command.
`EOC`	Specifies a command terminator. A command may be terminated by entering a semicolon, another command, or by repeating the last xy location entered with the command (if the command utilizes xy points).

Example

```
HELP ARCHIVE;
```

result:

```
ARChive [(+|-|#)desc[layer]] [:L] [:Nlvl] [volume] name [volume] EOC
```

Example

`HELP ;` (from inside process subsystem)

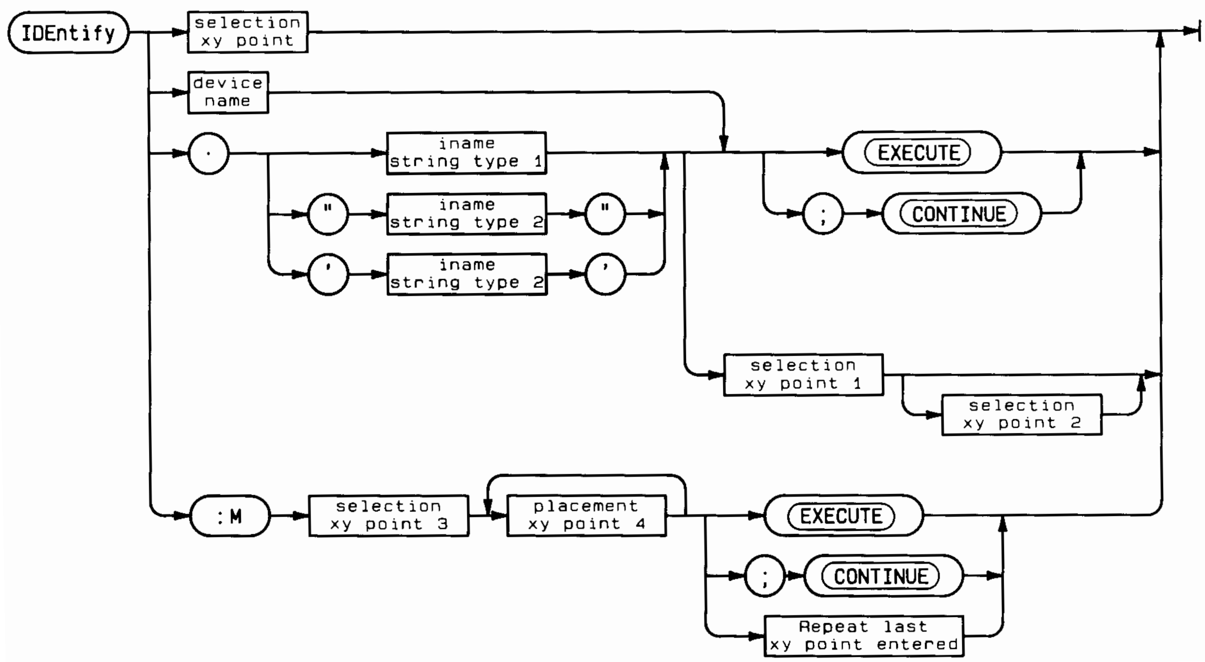
result:

```
Command syntax for PROC subsystem


SAVe [volume] (name | EOC)
LISt (PROcess|MACro|devicename|EOC)
HELp (commandname|macroname|EOC)
EXIt
ECHo (toggles on/off; On writes all input to printer)
EQUate [:(S|D|I)] [:Ccolor] [:Ppen] [:M(S|D|B|L)] [:T(I|T)[font]]
   [label] layer
UNIts (UM | CM | MIl | INch | (FT | FEet) ) resolution
REMove [volume] (name | EOC)
VOLumes EOC
PACk [volume] EOC
PREfix [volume] EOC
CATalog [volume] EOC
```

IDENTIFY

Valid in:
Edit subsystem

The IDENTIFY command allows you to: highlight all instances with the same device name, display the iname of a particular component, change the iname of a particular component, highlight all components with the same iname, and move the location of an iname relative to the component with which it is associated.



Item	Description/Default	Range Restrictions
device name	Name of the instance(s) that you wish to be highlighted on the display.	
iname string type 1	Set of characters forming the iname that you wish to display or to assign. All characters may be displayed and/or plotted.	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
iname string type 2	Set of characters forming the iname that you wish to display or to assign. All characters may be displayed, however only those characters common to string type 1 and string type 2 may be plotted.	legal characters: A-Z, 0-9, (<),(>), (/), (\), (%), (!), ( ), (,), (.), (;), (/), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?)), (=)
selection xy point 1	Xy point lying in or on the component. Entering xy point 1 causes the iname of the component to be displayed, if the component has an iname. If the component does not already have an iname, the iname specified with IDENTIFY command is assigned to the component.	
selection xy point 2	Xy point lying in or on the component. Entering this point assigns the iname entered with the IDENTIFY command to as the new iname of the component selected with xy point 1.	
selection xy point 3	Xy point lying in or on a component. Entering this point causes the iname assigned to the component to be displayed.	
selection xy point 4	Xy point lying in or on a component. This point identifies the component whose iname you wish to move.	
placement xy point	Xy point specifying the new location of the iname. The iname is moved such the lower left corner of the iname corresponds to the placement xy point. Entering additional placement points allows you replace the previous placement point with the new placement point, thus the placement point is a trial placement location.	

HELP syntax

```
IDEntify <xysel ... | devicename | .iname <xy1[xy2]|EOC> | :M xysel
[xypnt ...] EOC>
```

Use

The IDENTIFY command allows you to: highlight all instances with the same iname of a particular component, highlight all components with a particular iname, and move the location of an iname relative to the component with which it is associated.

Entering IDENTIFY with a device name causes all instances in the drawing with that device name to be highlighted in the display.

Entering IDENTIFY with an iname string allows you to assign the iname to the component selected. If the component already has an iname, the first xy selection point causes it to be displayed while the second xy selection point causes the old iname to be replaced with the new iname. If the component does not have an iname, the iname entered with the IDENTIFY command is assigned to the component when the first xy selection point is entered.

Entering the IDENTIFY command with only a selection point causes the iname of the component selected to be displayed. In addition, all attributes of the component selected (such as layer number, font size, scaling and rotation) are displayed at the top of the screen.

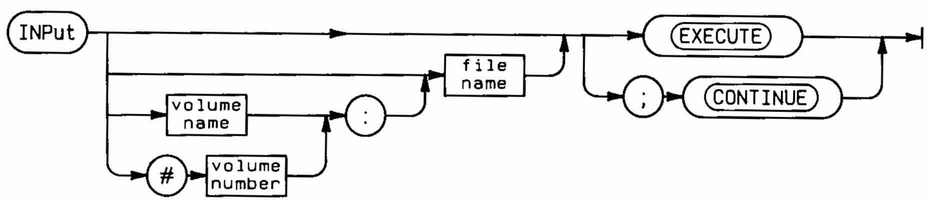
Entering the IDENTIFY command with the :M option allows you to move the location of the iname assigned to a component. To move the iname, select a point that lies in or on the component whose iname you wish to move. Then select the xy location where you wish the iname to be plotted on the display.

When a component is selected with the IDENTIFY command, the component type and the layer on which it is entered are displayed in the upper left corner of the CRT.

INPUT

Valid in:
Edit subsystem

The INPUT command allows you to recover the device file from which a generate file was created.



Item	Description/Default	Range Restrictions
volume name	Name of the volume containing the generate file from which you wish to re-construct a device.	
volume number	Number of the volume containing the generate file from which you which to reconstruct a device.	3-50
file name	Name of the generate file from which you wish to reconstruct a device.	

HELP syntax

```
INPut {[[volume] filename] | EOC}
```

Use

Enter the INPUT command along with the name of the generate file. The system then begins re-constructing the device from the information contained in the generate file. As each command is read from the generate file, it is executed in the edit subsystem, thus recreating the original drawing.

Simultaneously pressing the SHIFT key and the PAUSE key during the input process allows you to abort the re-construction process or to make additional inputs to the device via the computer keyboard or the graphic input device. When these keys are pressed, the system prompts you to see if you really desire to abort the input process. Answering Y, for yes, causes the system to abort the input operation and returns control to the keyboard or graphic input device. Answering N, for no, suspends (not aborts) the input process and allows you to enter commands from the keyboard or the graphic input device to make additions/changes to the device being created. To re-enable the input process, enter:

```
INPUT ;
```

and the input process continues from the point where the abort was requested.

If an error occurs during the input process, (such as an improper command in the generate file), and echo is disabled, input switches to the keyboard or to the graphic input device and the input process is suspended. You may correct the error by entering a correct version of the erroneous command on either the keyboard or the graphic input device. Entering:

```
INPUT ;
```

causes the input process to be resumed.

If an error occurs during the input process and echo is enabled, the error is printed on the system's built in printer and the input process continues. You may then use the commands of the edit subsystem to correct the error when the input process is finished.

Example:

```
INPUT EXAMP ;
```

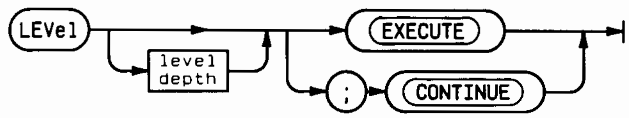
result:

The generate file named EXAMP located on the current system volume is read by the system. The commands forming the device from which the generate file was created are executed in the edit subsystem, and the components forming that original device are displayed on the CRT.

LEVEL

Valid in:
Edit subsystem

The LEVEL command is used to assign the logical level number to a drawing.



Item	Description/Default	Range Restrictions
level depth	The LEVEL depth to set the current drawing to. Default is 1.	−32767 to 32767

HELP syntax

```
LEVel {logic_level | EOC}
```

Use

The LEVEL command allows you to display or change the level number of the current drawing which is being edited. The default logical level is 1, and is assigned to a drawing when it is being created.

Whenever you are adding instances to a drawing, the level of the instance being added must be less than or equal to the level number of the drawing.

The primary use of the LEVEL command is during a generate of a drawing. The GENERATE command has an option (:D) which allows you to specify the level depth to which a drawing is to be generated. If (:D) is not specified, then the logical levels are ignored during the generation process.

Example:

```
LEVEL 5 ;
```

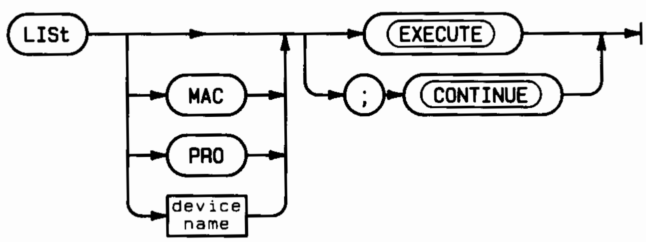
result:

Sets the current LEVEL depth to 5.

LIST

Valid in:
Graphics Editor
Edit subsystem
Macro subsystem
Process subsystem

The LIST command provides a listing of the macro, process, and menu files currently loaded into the system as well as a listing of all devices currently in computer memory. Additionally, you can obtain a list describing any particular device resident in computer memory. This list specifies how many of each type of component is used in the device as well as a list of all devices linked to that device.



Item	Description/Default
PRO	Specifies to list the contents of the current process file.
MAC	Specifies to list the contents of the current macro file.
device name	Name of the device you want to list.

HELP syntax

```
LISt (PROcess|MACro|devicename|EOC)
```

Use

Entering the LIST command from the Graphics Editor causes the system to print a listing of the menu, macro, and process files currently loaded into the system. Additionally, a list of all devices currently resident in computer memory is provided.

Entering the LIST command from the edit subsystem causes the system to print a listing of the following items: the current device name, lock angle, window parameters, grid setting, number of each component type used, the links to other devices (instance components), and the number of records that the drawing occupies in the computer memory.

Entering the LIST command from the process subsystem causes the system to print a listing of the contents of the current process file.

Entering the LIST command from the macro subsystem causes the system to print a listing of the contents of the current macro file.

Example:

`LIST;` (from inside the edit subsystem)

result:

```
Device file: EWSYS:EXAMP

Resolution:   1 UM              is equal to 1 system grid point.

                                           X          Y     UM

Boundary rectangle: upper right     190.0     210.0
                    lower left     -190.0    -190.0

Grid setting:      Size      Multiple
                   10.0          1     0.0,      0.0

Lock Angle =  15.00 degrees

Component Record Data          # of Records

Primitive type
Rectangl                             3
Line                                 2
Circle                               1
Arc                                  1
Text                                 1
                               --------
     Total                           8
Deleted components                   0
                               --------
Total records used:                  8

Unused memory = 276 Kbytes
```

Example:

`LIST PRO;` (from inside the edit subsystem)

result:

```
Resolution:   1 UM              is equal to 1 system grid point.

Process file is EWSYS:PROCDATA
```

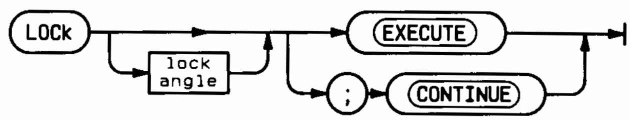
LAYER #	PEN #	LABEL	COLOR	LINE TYPE	LAYER TYPE	IDENTIFY NAME disp	IDENTIFY NAME size	TRACE NAME disp	TRACE NAME size
0	0	INSTANCE	White	Dotted	Boundary	OFF	1	OFF	1
1	1		Red	Solid	Detail	OFF	1	OFF	1
2	2		Yellow	Solid	Detail	OFF	1	OFF	1
3	3		Green	Solid	Detail	OFF	1	OFF	1
4	4		Aqua	Solid	Detail	OFF	1	OFF	1
5	5		Blue	Solid	Detail	OFF	1	OFF	1
6	6		Purple	Solid	Detail	OFF	1	OFF	1
7	1		Red	Dotted	Detail	OFF	1	OFF	1
8	2		Yellow	Dotted	Detail	OFF	1	OFF	1
9	3		Green	Dotted	Detail	OFF	1	OFF	1
10	4		Aqua	Dotted	Detail	OFF	1	OFF	1
11	5		Blue	Dotted	Detail	OFF	1	OFF	1
12	6		Purple	Dotted	Detail	OFF	1	OFF	1
13	1		Red	Broken	Detail	OFF	1	OFF	1
14	2		Yellow	Broken	Detail	OFF	1	OFF	1
15	3		Green	Broken	Detail	OFF	1	OFF	1
16	4		Aqua	Broken	Detail	OFF	1	OFF	1
17	5		Blue	Broken	Detail	OFF	1	OFF	1
18	6		Purple	Broken	Detail	OFF	1	OFF	1
19	1		Red	Dashed	Detail	OFF	1	OFF	1
20	2		Yellow	Dashed	Detail	OFF	1	OFF	1
21	3		Green	Dashed	Detail	OFF	1	OFF	1
22	4		Aqua	Dashed	Detail	OFF	1	OFF	1
23	5		Blue	Dashed	Detail	OFF	1	OFF	1
24	6		Purple	Dashed	Detail	OFF	1	OFF	1
30	1		Red	Solid	Detail	ON	10	ON	10
31	2		Yellow	Solid	Detail	ON	10	ON	10
32	3		Green	Solid	Detail	ON	10	ON	10
33	4		Aqua	Solid	Detail	ON	10	ON	10
34	5		Blue	Solid	Detail	ON	10	ON	10
35	6		Purple	Solid	Detail	ON	10	ON	10
40	1	INTER1	Red	Solid	Interconnect	OFF	1	OFF	1
41	2	INTER2	Yellow	Solid	Interconnect	OFF	1	OFF	1
42	3	INTER3	Green	Solid	Interconnect	OFF	1	OFF	1
43	4	INTER4	Aqua	Solid	Interconnect	OFF	1	OFF	1
44	5	INTER5	Blue	Solid	Interconnect	OFF	1	OFF	1
45	6	INTER6	Purple	Solid	Interconnect	OFF	1	OFF	1
50	1	SYMB1	Red	Solid	Symbolic	OFF	1	OFF	1
51	2	SYMB2	Yellow	Solid	Symbolic	OFF	1	OFF	1
52	3	SYMB3	Green	Solid	Symbolic	OFF	1	OFF	1
53	4	SYMB4	Aqua	Solid	Symbolic	OFF	1	OFF	1
54	5	SYMB5	Blue	Solid	Symbolic	OFF	1	OFF	1
55	6	SYMB6	Purple	Solid	Symbolic	OFF	1	OFF	1

```
Unused memory = 276 Kbytes
```

LOCK

Valid in:
Edit subsystem

The LOCK command determines the default resolution for circle and arc components. Additionally, the LOCK command allows you to set the angle for the lock angle warning.



Item	Description/Default	Range Restrictions
lock angle	The lock angle, in degrees, that you wish to set. The value of the lock angle when the edit subsystem is first entered is 15 degrees.	0 - 90.00 in .01 increments

HELP syntax

```
LOCk <angle | EOC>
```

Use

The LOCK command determines the default resolution for circle and arc components. Additionally, it sets the angle for the lock angle warning.

The lock angle warning is a warning issued by the system when the angle of a component (such as the angle of a line or of a note) is not an integer multiple of the lock angle.

This warning is not a critical one, rather it is a message to aid you when entering components. For example, when laying out traces for a printed circuit board, the angle formed by the intersection of two traces (line segment) is usually 45 degrees. The lock angle warning could notify you whenever the angle is not exactly 45 degrees (or an integer multiple of 45).

Entering `LOCK ;` causes the system to display the current lock angle.

Example:

```
LOCK 30 ;
```

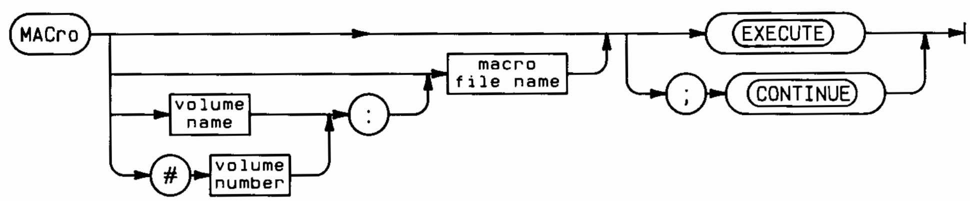
result:

Sets the current lock angle at 30.00 degrees.

MACRO

Valid in:
Graphics Editor

The MACRO command allows you to load a particular macro file and provides access to the macro subsystem. Once in the macro subsystem, you may use any of the commands valid there to modify an existing macro file, create new macro commands, or create a new macro file.



Item	Description/Default	Range Restrictions
volume name	Name of the volume where you wish to get the macro file.	
volume number	Number of the volume where you wish to get the macro file.	3-50
macro file name	Name of the file to be loaded. The first character of the file name must be a letter (A-Z) or the underscore character (_).	8 characters max legal characters: A-Z, 0-9, (_)

HELP syntax:

```
MACro [volume] {filename | EOC}
```

Use

Entering the MACRO command with and without a file name can cause varied results.

- If no file name is supplied, the system uses the macro file currently loaded into the system and no new file is loaded.

  If no macro file is currently loaded and no file name is supplied with the MACRO command, no file is loaded; however, a default macro file name, MACRDATA, is assigned as the default macro file name.

- If a file name is supplied, the result depends upon whether a source volume specifier is entered and upon whether the macro file exists.

  If a source volume specifier is supplied, the system searches only the volume specified for the macro file. If no source volume specifier is supplied, the system searches all volume currently on line, beginning with the system volume, EWSYS, until the macro file is found.

  If a file name is supplied and a macro file with that name exists, that macro file is loaded, thus becoming the current macro file.

  If a file name is supplied and a macro file with that name is not found, the current macro file remains; however, the new macro file name is assigned as the default macro file name.

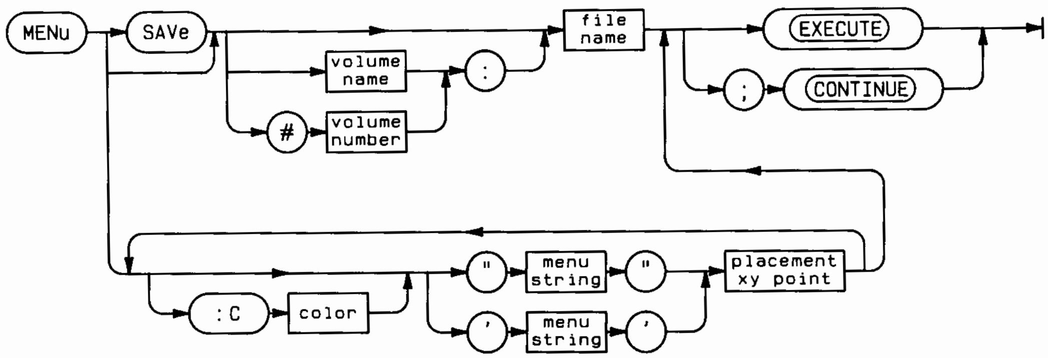
Examples:

```
MACRO ;
MACRO #3:MACRDATA
MACRO EWSYS:MACRDATA
```

MENU

Valid in:
Edit subsystem

The MENU command allows you to load a different menu into the menu area of the display, modify the existing menu by adding new entries to the menu or deleting entries from the menu, and store the current menu in a menu file.



Item	Description/Default	Range Restrictions
SAVE	Specifies that you desire to save the current menu configuration in a menu file.	
menu name	Name of the menu file to be loaded into the system or the name you wish to assign to the menu being saved. The default menu name is the name of the current menu file.	Cannot be the name of an existing system command or valid macro
volume name	Name of the volume containing the menu file.	
volume number	Number of the volume containing the menu file.	3-50
color	Single character specifying the color to display the string being added to the current menu (systems with color CRTs only). The default color is green.	A = aqua B = blue G = green P = purple R = red Y = yellow W = white
menu string	Set of characters (8 maximum) that you wish to add to the current menu.	Any alphanumeric character except square brackets, [ ]
placement point	Xy location in the menu specifying the menu box where the menu string is to be added. This point must be selected with the graphic input device.	

HELP syntax

```
MENU {[:Ccolor] "entry" menusel | [SAVE] [volume]{file|EOC}}
```

Use

The MENU command allows you to: load a different menu into the menu area of the display, modify the existing menu by adding new entries to the menu or deleting entries from the menu, and store the current menu in a menu file.

New entries may be made to the menu area by entering the MENU command along with a string of up to eight characters. Then select a point inside the menu box where you wish the entry to be placed. If that menu space is already occupied, the new menu item replaces the old. By entering an empty string (`MENU "  "`) in a menu box you can effectively erase an existing entry, since the existing entry is replaced with the new entry (a string of blank characters).

If a macro name is being added to a menu slot and the macro name is 7 characters or less, a trailing blank must be added to the menu entry. For example, to add the macro name ARCS to a menu slot, you should enter `MENU "ARCS "`(the characters ARCS plus 1 blank).

Example:

```
MENU SAVE EW1:BOB ;
```

result:

The current menu is saved in the menu file BOB on the volume EW1.

Example:

```
MENU :CR "AREA" xy point ;
```

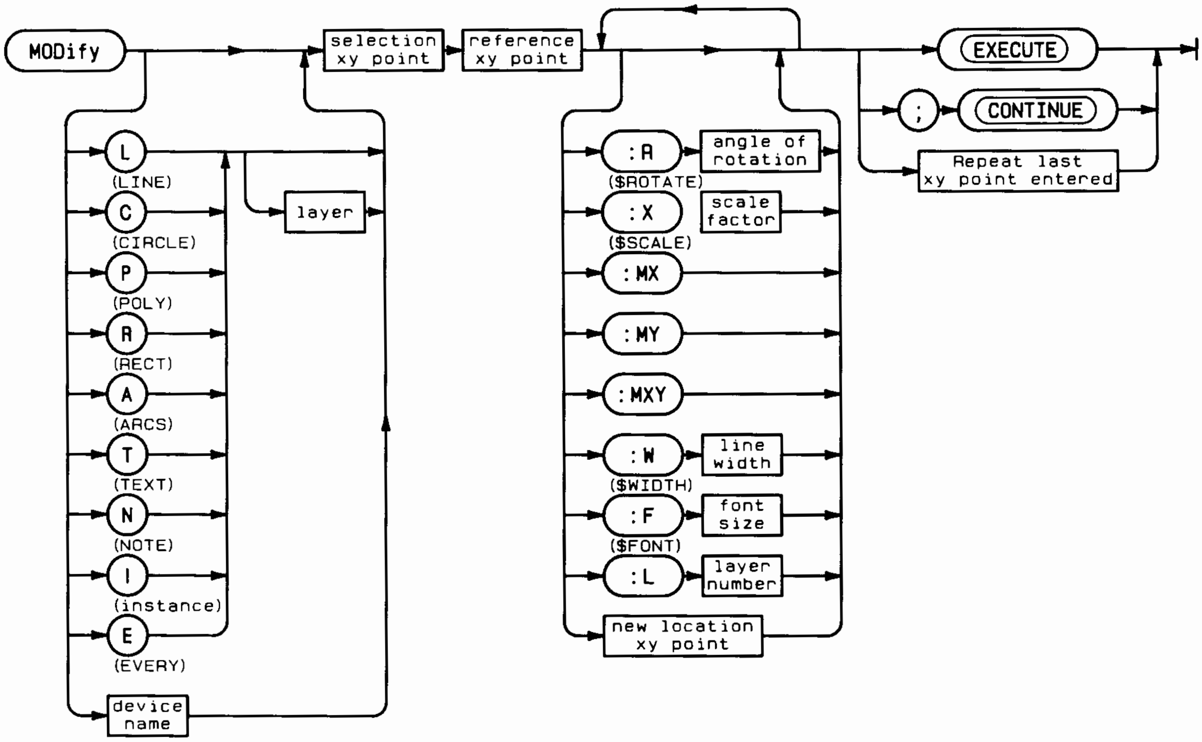
result:

The characters AREA are added to the menu in the menu box specified by xy point. The characters are displayed in red.

MODIFY

Valid in:
Edit subsystem

The MODIFY command allows you to change existing components on the screen. These changes include rotating, mirroring, scaling, font size, line width, and changing layers.



Item	Description/Default	Range Restrictions
A, C, E, I, L, N, P, R, T,	Limits selection of the component to be modified to the component type specified (such as R for rectangle).	
device name	Name of the instance to be modified.	
layer	Layer containing the component to be modified.	1-255
selection xy point	Point lying in or on the component to be modified.	
reference point	Xy point identifying the part of the component to be modified.	
angle of rotation	Angle, in degrees, which you wish to rotate the component. The component is displayed at the rotated angle.	– 180.00 to 180.00

scale factor	Amount by which you wish to change the scale of the existing component when displayed in your drawing. The current units of the component are multiplied by the scale factor to obtain the new scale. For example, entering a scale factor of 5 causes the dimensions of the component to be one half of the original dimensions of the component. Entering a negative scale factor causes the scale of the component to be divided by the scale factor, instead of multiplied. For example, entering the scale factor -2, causes the dimensions of the component to be one half of the original dimensions.	
:MX	Specifies that the component to be modified is to be mirrored about the X axis (see the drawings below).	
:MY	Specifies that the component to be modified is to be mirrored about the Y axis (see the drawings below).	
:MXY	Specifies that the component to be modified is to be mirrored about both the X and the Y axes.	
line width	Width of the line in current user units. (line components only)	width >0 maximum width is the number of user units assigned to 16 383 system grid points.
font size	Height of the characters, in current units, from which the text is to be formed. The maximum font size is specified by the number of user units assigned to 16 383 system grid points (a very large font size).	1 - (number of user units assigned to 16 383 system grid points)
layer #	The new layer number for the component selected. Only instance components cannot be moved to a new layer.	1-255
new location point	Xy point identifying the location to which you want to move the component. The component is moved such that the reference point coincides with the new location point. Entering additional xy points replaces the location point with each new point entered. Thus, the location point is a trial placement location.	

HELP syntax:

```
MODify [{desc[layer]|devicename}] xysel xyref [[:Rrot] [:Xscl]
[:M[X|Y|XY] [:Wwidth] [:Ffont] [:Llayer] [xy1] ...]] EOC
```

Use

The MODIFY command allows you to change existing components of the current drawing. However, modifications include rescaling, rotation, mirroring about the X and/or Y axis, changing the width of lines, changing the font size of notes and text, and changing the layer number on which the selected component resides. Additionally, you may change the position of the component within the drawing, as with the MOVE command.

All of the modifications are interactive; the results are displayed on the CRT as they happen. Executing the UNDO command following the MODIFY command returns the modified component to its original state.

When modifying the scale, rotation, or mirroring of a component, the modification is done relative to the XY reference point specified. These three operations are always performed relative to the original component when the component is selected. For example, when a rotation of 45 degrees is entered, followed by another rotation of 50 degrees, the component will be displayed at 50 degrees (not 95 which is additive).

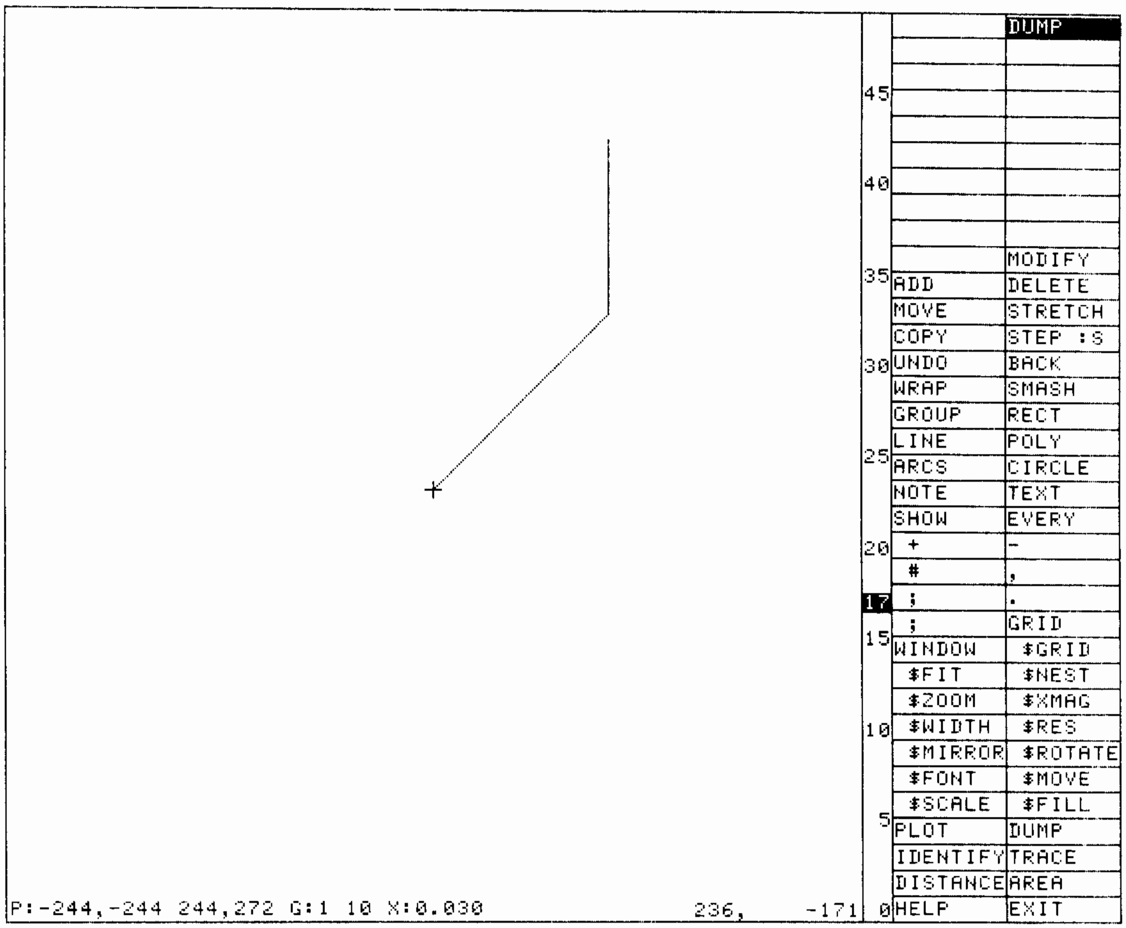
Note

When entering values via the numeric column, use a delimiter (comma, space, or CONT) or another command to terminate the value. Numbers entered via the column can be entered intwo steps (i.e., 75) which means that the system needs a delimiter to know when input from the column has finished.

As the parameters for a particular component are changed, the information in the display area reflects that change. However, when changing layer number, the color and/or line type will not change until the command is finished.

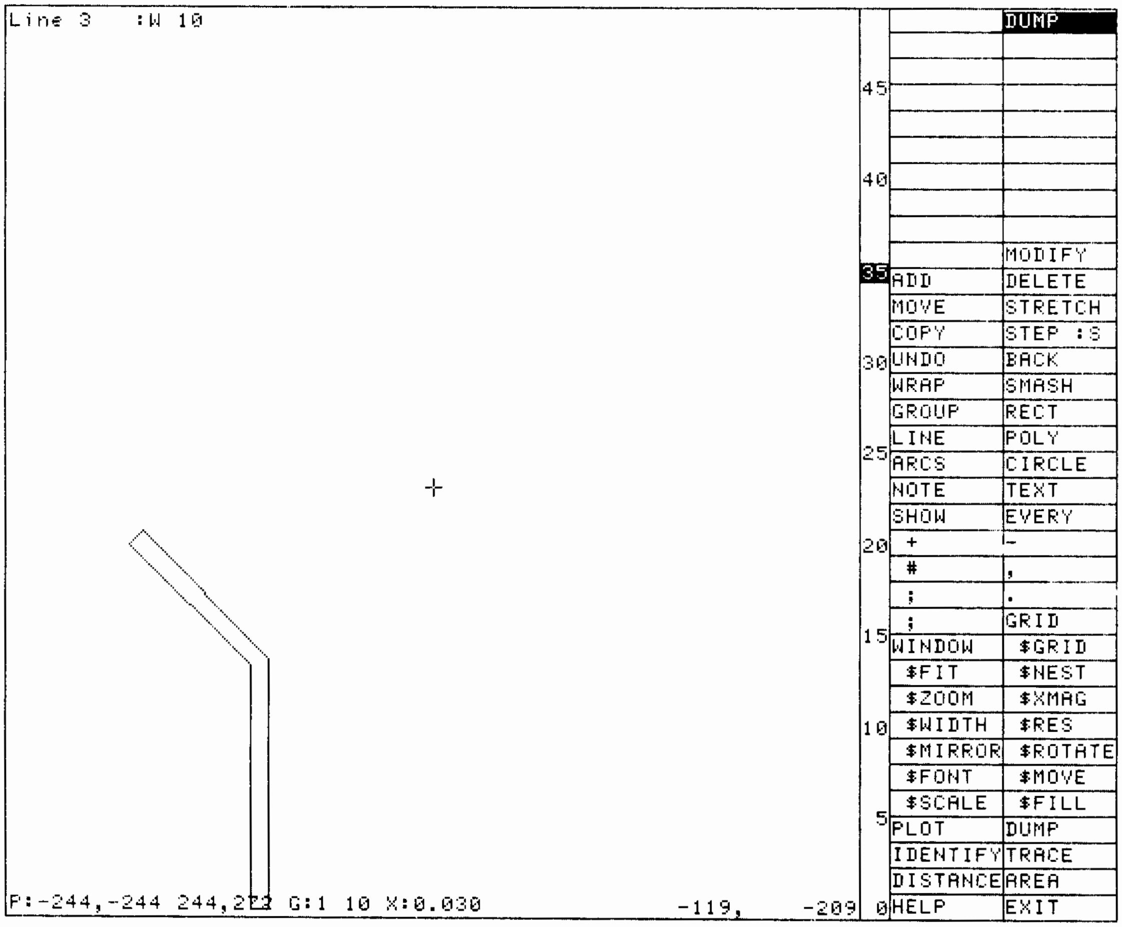
Example:
Add a line to the display on layer #4 with vertices at 0,0 100,100 and 100,200.

```
ADD LINE 4 0,0 100,100 100,200;
```



Modify the line which has just been drawn. Set the origin at 100,100, rotate the line 45 degrees, give the line a width of 10, change the layer number 3 and move the line to −100,100.

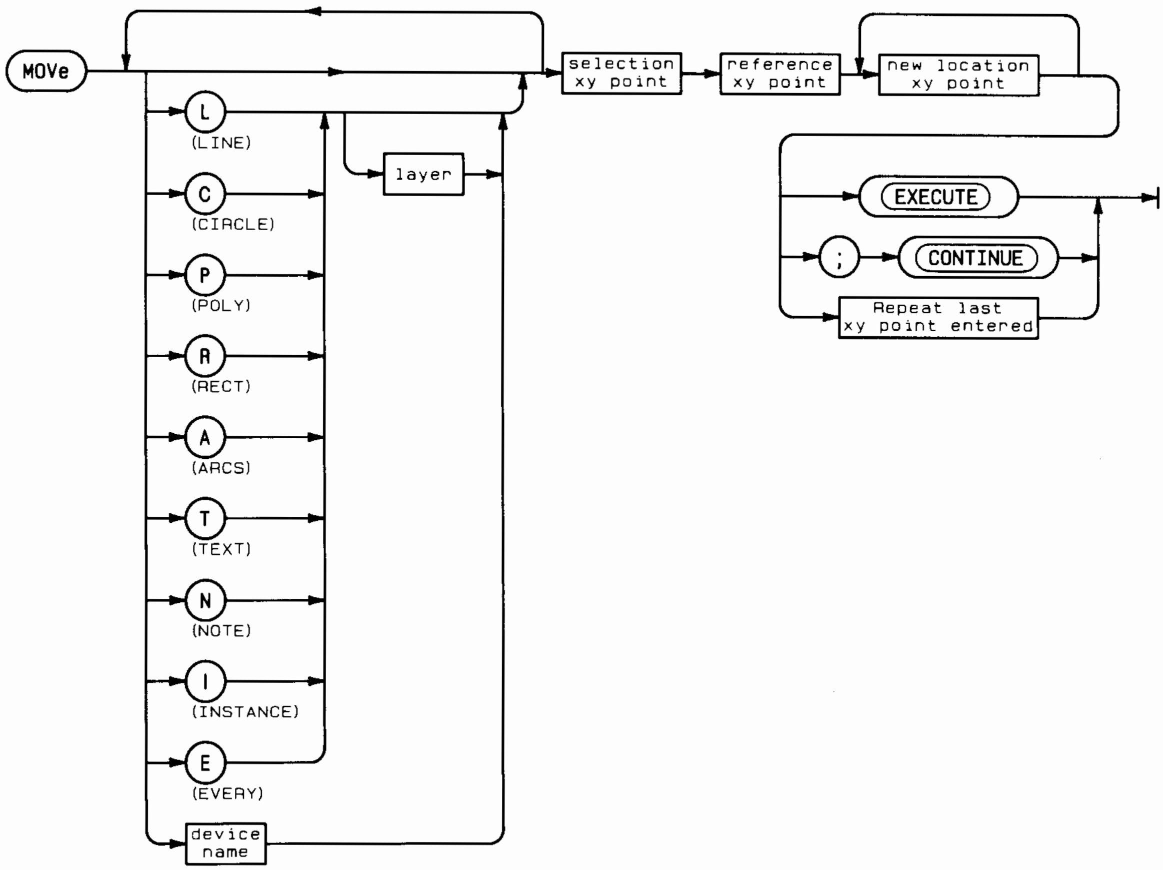
```
MODIFY 0,0 100,100 :R 45 :W10 :L3 -100,100;
```



MOVE

Valid in:
Edit subsystem

The MOVE command allows you move any component from its current location to a new location on the drawing.



Item	Description/Default	Range Restrictions
A, C, E, I, L, N, P, R, T,	Limits selection of the component to be moved to the component type specified (such as R for rectangle).	
layer	Layer containing the component to be moved.	1 - 255
device name	Device name of the instance that you wish to move.	
selection xy point	Point lying in or on the component to be moved.	
reference point	Xy point reference point used to accurately place the component being moved.	
new location point	Xy point identifying the location to which you want to move the component. The component is moved such that the reference point coincides with the new location point. Entering additional xy points replaces the location point with each new point entered. Thus, the location point is a trial placement location.	

HELP syntax

```
MOVe [(desc[layer]|devicename)] xysel xyref xy1 [xy2...] EOC
```

Use

The component to be moved is selected by specifying a point in or on the component. The layer and descriptor (or device name, for instance components) may be used to limit component selection to a specific type of component. A reference point is specified after the component is selected to aid in accurate placement at the new location. The component is moved such that the reference point coincides with the new location point.

Example:

```
MOVE T2 10,20 0,0 100,100 ;
```

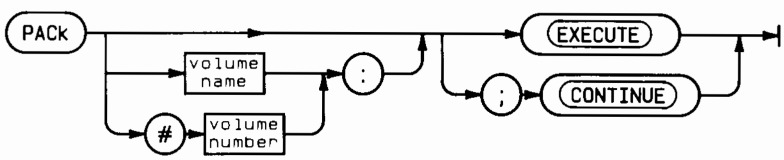
result:

A text component in layer 2, containing the location 10,20, is selected as the component to be moved. It is assigned a reference location at 0,0 and is moved such that the reference location coincides with the location, 100,100.

PACK

Valid in:
Graphics Editor
Edit subsystem
Macro subsystem
Process subsystem

The PACK command is used to remove all gaps left between existing files. This packs all files into the beginning of the volume, leaving the remaining "free space" in one large area at the end of the volume.



Item	Description/Default	Range Restrictions
volume name	Name of the volume you wish to pack.	6 characters max.
volume number	Number of the volume you wish to pack.	3-50

HELP syntax:

```
PACK [volume] EOC
```

Use

This command packs the specified volume such that the free (available) space is at the end of the volume. During daily operation, the volume becomes fragmented. Thus there are sections of unused disc area at different parts of the volume. As a result, there are times when there is enough total room on the volume to store a particular file, but not enough room in any one single place.

The PACK command recitifes this problem by, in effect, moving all free space to the end of the volume.

CAUTION

If there is a power failure during the pack command, data may be lost. It is recommended that you backup the volume before packing it.

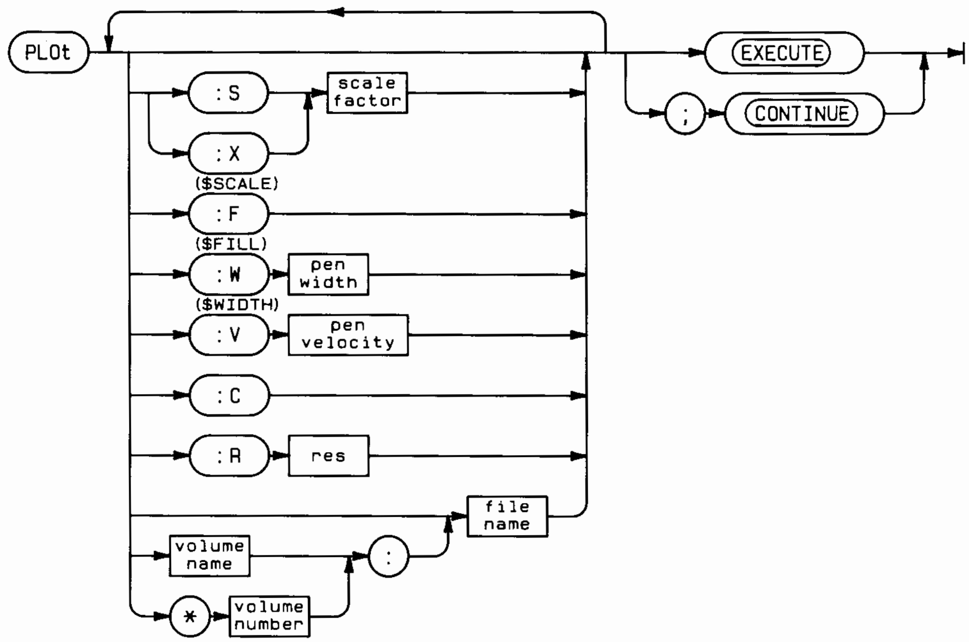
Examples:

```
PACK #3:;
PACK EWSYS:;
```

PLOT

Valid in:
Edit subsystem

The PLOT command allows you to obtain a pen plot of any drawing displayed on the CRT.



Item	Description/Default	Range Restrictions
:S or :X	Specifies to scale the drawing to be plotted. When a scale option is entered (either :X or :S), the drawing is plotted such that distance between points in user units, is plotted to the exact size indicated by those units. For example, if user units specify that the distance between endpoints of a line component is 5 inches, the line is plotted 5 inches long assuming that the scale factor equals 1).	
scale factor	Allows you to change the scale with which the drawing is plotted. The scale factor entered is multiplied by the x and y dimensions of each component before it is plotted. Entering the scale factor 2, causes a rectangle originally drawn 2" by 4" in user units, to be plotted 4" by 8".	
:F	Specifies to fill in solid all rectangles, circles, and lines (with width) when the drawing is plotted.	
pen width	Width of the plotter pen, in mils. This information is used to insure that when a component is plotted, the dimensions of the component are not changed because of the width of the pen used to plot the component. Additionally, it insures proper filling of components when the :F option is entered. The default pen width is 15 mils.	

Item	Description/Default	Range Restrictions
pen velocity	Velocity of the plotter pen in centimetres/second.	pen velocity must be an integer. HP 7225: 1-25 cm/sec HP 7580/7585 1-60 cm/sec HP 9872 B/C/T: 1-36 cm/sec HP 7470 1-36 cm/sec
:C	Specifies that the outer limits of the drawing are to be fit onto the plotter between P1 and P2 and centered. If scaling is active, then the plot will be performed only if there is sufficient room.	
curve resolution	Specifies that the resolution for all arcs and circles are to be overridden with the resolution (res) specified. If no resolution value is given, then the default resolution is 1 degree.	
file name	The name of the output file to send the plotter commands to rather than the plotter.	

HELP syntax

```
PLOt [:Sscl|:Xscl] [[volume]file_name] [:F] [:C] [:R[res]]
[:Wwid][:Vvelocity] EOC
```

Use

Entering the PLOT command causes the drawing currently displayed on the CRT to be plotted on the current plotter (as defined in the configuration file). The drawing is copied to the plotter with the lower left hand corner of the window copied to the defined lower left hand corner (P1) of the plotter.

If the scale option is not used, only the parts of the drawing visible in the drawing area are copied to the plotter. It is plotted such that the area visible in the drawing area is mapped to the area defined by P1 and P2 of the plotter. The units of the drawing are ignored.

If the user wishes, plotter output may be sent to a file. This file will contain the HPGL commands which are normally sent to the plotter. The system always assumes 8.5" by 11" paper when plotting to a file.

If the scale option is used, the drawing is plotted multiplied by the supplied scale factor. If a line in your drawing is defined to be 2 feet long in user units of the drawing and a scale factor of 3 is entered, the system attempts to plot that line 6 feet long.

If the scale option is used, all portions of the drawing lying in both the positive X direction and the positive Y direction from the lower left hand corner of the CRT are plotted (whether in the current display window or not). Portions of the drawing that cannot fit in the plotter area defined by P1 and P2 are not plotted.

The centering option (:C) will allow drawings outer limits to be fit and centered between P1 and P2 of the plotter. If scaling is specified, the drawing will be centered and drawn at the specified scale. If the combination of scaling and centering will not fit between P1 and P2 of the plotter, the system will abort the plot command and issue an error message, telling you what the maximum scale may be in order to center the drawing.

Example:

```
PLOT :S.5 :V16 ;
```

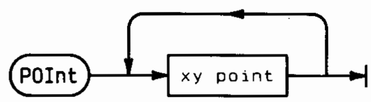
result:

A copy of the drawing currently displayed in the drawing area is plotted on the current plotter. The drawing is plotted to .5 times full scale, as defined by the user units of the drawing. The plot is plotted at 16 centimetres per second.

POINT

Valid in:
Edit subsystem

The POINT command allows you to display the xy coordinates of the specified point, in user units.



Item	Description
xy point	Location where you wish to find the xy coordinates.

HELP syntax

```
POInt xypnt
```

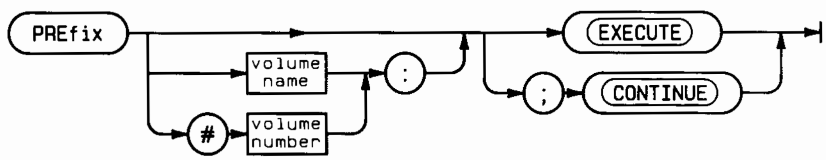
Use

Enter the POINT command and select the location on the CRT you wish to know the xy coordinates of. This information is erased when the screen is redrawn.

PREFIX

Valid in:
Graphics Editor
Edit subsystem
Macro subsystem
Process subsystem

The PREFIX function allows you to change the default volume used in the Graphics Editor and its subsystems. Whenever a volume is not explicitly or implicitly stated, the Graphics Editor uses the default volume.



Item	Description/Default	Range Restrictions
volume name	Name of the volume you wish to set as the prefix.	6 characters max
volume number	Number of the volumes you wish to set as the prefix	3-50

HELP syntax

```
PREfix [volume] EOC
```

Use

The Prefix function allows you to change the default volume. This is valid only in the graphics editor. Whenever a volume is not explicitly or implicitly stated, the system uses the default volume as the volume to be acted upon. The default volume is the system volume, EWSYS, when the graphics editor is first selected. If no volume is specified, the current prefix volume is displayed.

Note

When attempting to bring a new file into the system and a volume is not specified, the graphics editor first searches the prefix volume for the file. If the file range is not found there, the system searches all other volumes on line until the file is found.

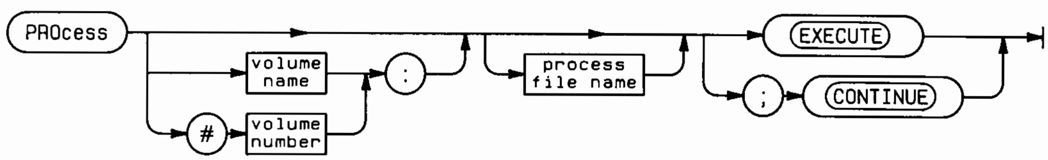
Example:

```
PREFIX EWSCH: ;
PREFIX #4: ;
```

PROCESS

Valid in:
Graphics Editor

The PROCESS command allows you to load a particular process file and provides access to the process subsystem. Once in the process subsystem, you may use any of the commands valid there to modify an existing process file, change the current layer definitions, or create a new process file.



Item	Description/Default	Range Restrictions
volume name	Name of the volume where you wish to get the process file.	
volume number	Number of the volume where you wish to get the process file.	3-50
file name	Name of the process file to be loaded. The first character of the file name must be a letter (A-Z) or the underscore character (_).	legal characters: A-Z, 0-9, (_)

HELP syntax

```
PROcess [volume] <filename | EOC>
```

Use

Entering the PROCESS command with and without a file name can cause varied results.

- If no file name is supplied, the system uses the process file currently loaded into the system and no new file is loaded.

  If no process file is currently loaded and no file name is supplied with the PROCESS command, no file is loaded; however, a default process file name, PROCDATA, is assigned as the default process file name.

- If a file name is supplied, the result depends upon whether a source volume specifier is entered and upon whether the process file exits.

  If a source volume specifier is supplied, the system searches only the volume specified for the process file. If no source volume specifier is supplied, the system searches all volumes currently on line, beginning with the system volume, EWSYS, until the process file is found.

  If a file name is supplied and a process file with that name is found, that process file is loaded, thus becoming the current process file.

  If a file name is supplied and a process file with that name is not found, the current process file remains; however, the new process file name is assigned as the default process file name.

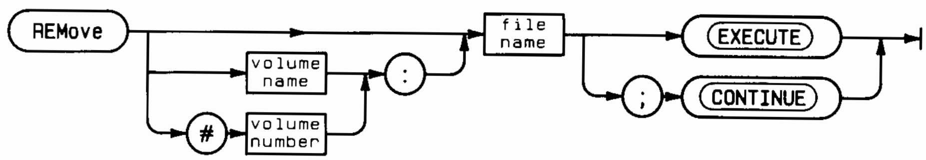
Examples:

```
PROCESS;
PROCESS EWSYS:PROCDATA;
PROCESS #3:PROCDATA;
```

REMOVE

Valid in:
Edit subsystem
Macro subsystem
Process subsystem

The REMOVE command allows the user to delete files from a volume.



Item	Description/Default	Range Restrictions
volume name	Name of the volume where the file is located.	6 characters max
volume number	Number of the volume where the file is located.	3-50
file name	Name of the file to be removed. The first character of the file name must be a letter (A-Z) or the underscore character (_).	8 characters max legal characters: A-Z, 0-9, (_)

HELP syntax

```
REMove [volume] {name | EOC}
```

Use

This command allows you to remove files from any volume currently on line. The command may be used in the edit, macro or process subsystem. Only those files which are characteristic to the subsystem you are in may be removed. For example, only process files may be removed while you are in the process subsystem. Menu files may not be deleted using the REMOVE command. The EWE filer program may be used to remove any file.

The system always requests the verification before the file is actually removed. If you answer yes, the file is removed; if you answer no, the remove command is aborted.

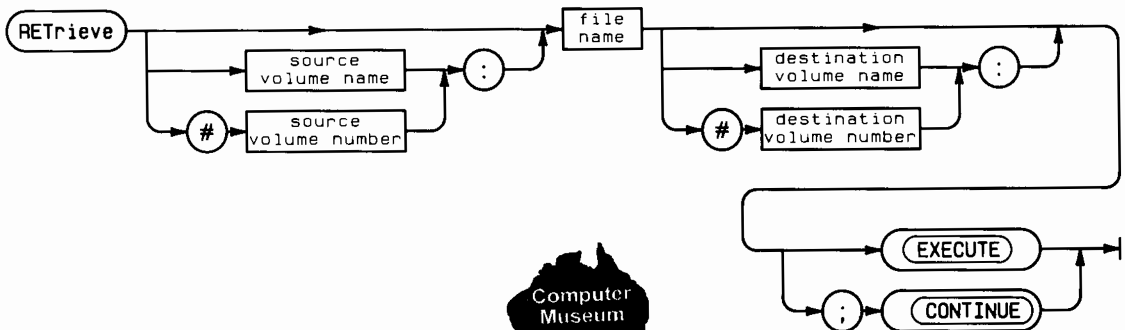
Note

If a device file exists both in the computer's memory and on disc, the graphics editor prompts you twice, once for memory removal and once for file removal.

RETRIEVE

Valid in:
Graphics Editor

The RETRIEVE command allows you to reconstruct a device file from an archive file.



Item	Description/Default	Range Restrictions
Source volume name	Name of the volume containing the archive file to be loaded.	
Source volume number	Number of the volume containing the archive file to be loaded.	3-50
file name	Name of the archive file from which you wish to construct a device file.	
destination volume name	Name of the volume where you wish to store retrieved device data.	
destination volume number	Number of the volume where you wish to store retrieved device data.	3-50

HELP syntax:

```
RETrieve <[volume] filename [volume] | EOC>
```

Use

The RETRIEVE command allows you to create a device from the contents of an archive file. This is the reciprocal function of the ARCHIVE command. Entering the command causes the system to read the contents of the archive file and line by line reconstruct the device from which the archive file was created. If the device contains links to other devices, these linked devices are also reconstructed.

Entering the RETRIEVE command causes the system to enter the edit subsystem to create the device and any linked devices. As the device is created, the individual edit subsystem commands stored in the archive file are entered. The results are displayed on the CRT just as if the commands were being entered interactively. Once the device is reconstructed, the system saves the device(s) and places the device file(s) on the specified volume.

You may stop the retrieve process at any time and either abort the command or make additional entries to the drawing. This is done by entering one or more edit subsystem commands. The retrieve process may then be reactivated from the point at which it was interrupted. This is done by entering the INPUT command.

If the retrieve process is aborted, you are left in the edit subsystem with the partially reconstructed device displayed on the CRT. You may then use the commands of the edit subsystem to modify or save the device. Additionally, you may exit the edit subsystem without saving the device by entering the EXIT command twice.

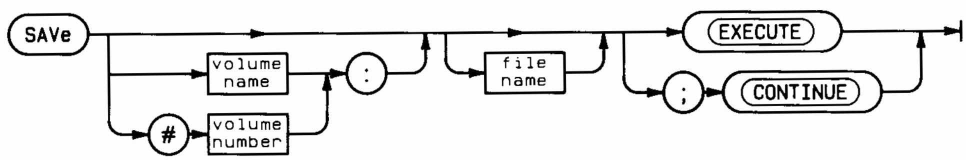
Entering the RETRIEVE command causes the loss of some existing device files. When retrieving an archive file, the system first removes any device file on the destination volume whose file name matches the name of the device. Additionally, if the reconstructed device is linked to other devices, the system removes all device files on the destination volume whose file name matches the name of the linked device(s). This allows the system to store the drawing and linked devices on the destination volume with the same device file names with which they were originally stored. This eliminates the possibility of creating duplicate device files.

Example:

```
RETRIEVE EWSYS:EXAMP;
RETRIEVE #3:EXAMP #4:;
RETRIEVE EWSYS:EXAMP EWSYS1:;
```

SAVE

Valid in:
Edit subsystem
Macro subsystem
Process subsystem



Item	Description/Default	Range Restrictions
volume name	Name of the volume where you wish to store the data	
volume number	Number of the volume where you wish to store the data.	3-50
file name	Name that you want to assign to the file in which the data is stored. The first character of the file name must be a letter (A-Z) or the underscore character (_).	8 characters max legal characters: A-Z, 0-9, (_)

HELP syntax

`SAVe [volume] {name | EOC}`

Use

EDIT Subsystem

The SAVE command allows you to store the drawing currently being edited and any named device linked to that drawing. If no device name is supplied with the SAVE command, the system uses the device name entered with the EDIT command when entering the edit subsystem as the device file name.

If no device name is supplied with the SAVE command and no device name was supplied with the EDIT command when entering the edit subsystem, an error message is displayed and the command must be re-entered with a device name supplied.

You cannot enter a name for the drawing that could be interpreted by the system to be a command name or a macro name. For example, entering `SAVE WIND` ... would cause the system to reject the SAVE command since WIND is a valid entry for the WINDOW command. Similarly, if a macro named ARCS is defined, entering `SAVE ARCS` ... would cause the system to reject the SAVE command.

When a drawing is saved, all named devices linked to that drawing that were created during the edit session are also saved.

To save only a named linked device: exit the edit subsystem, re-enter the edit subsystem with EDIT devicename (where devicename is the name of the device that you wish to save), and finally enter the SAVE command.

Example:

```
SAVE #3:EXAMP ;
SAVE EWSYS:EXAMP ;
```

result:

Both commands have the same result. The drawing currently displayed in the drawing area of the CRT is stored in a device file named EXAMP on the system volume, EWSYS.

PROCESS Subsystem

Entering the SAVE command allows you to store the layer definitions and the units definition in a special file called a process file.

If no file name is entered, the system uses the name of the process file that was loaded when you entered the process subsystem (i.e., the name supplied with the PROCESS command).

If no file name was supplied with the PROCESS command when entering the process subsystem and no file name is supplied with the SAVE command, the system saves the process file with the file name, PROCDATA.

You cannot use a file name that could be interpreted by the system to be a command name or a macro name. For example, entering `SAVE WIND` would cause the system to reject the SAVE command since WIND is a valid entry for the WINDOW command. Similarly, if a macro named ARCS is defined, entering `SAVE ARCS` will cause the system to reject the SAVE command.

MACRO Subsystem

Entering the SAVE command causes the set of macros currently defined to be stored in a special file, called a macro file. If no macro file name is supplied with the SAVE command, the system uses macro file name supplied with the MACRO command when entering the macro subsystem. This causes the macros stored in that file to be replaced with the macros in the new file.

If no macro file name is supplied with the SAVE command and no macro file name was used with the MACRO command when entering the macro subsystem, the system uses the file name, MACRDATA to store the macros.

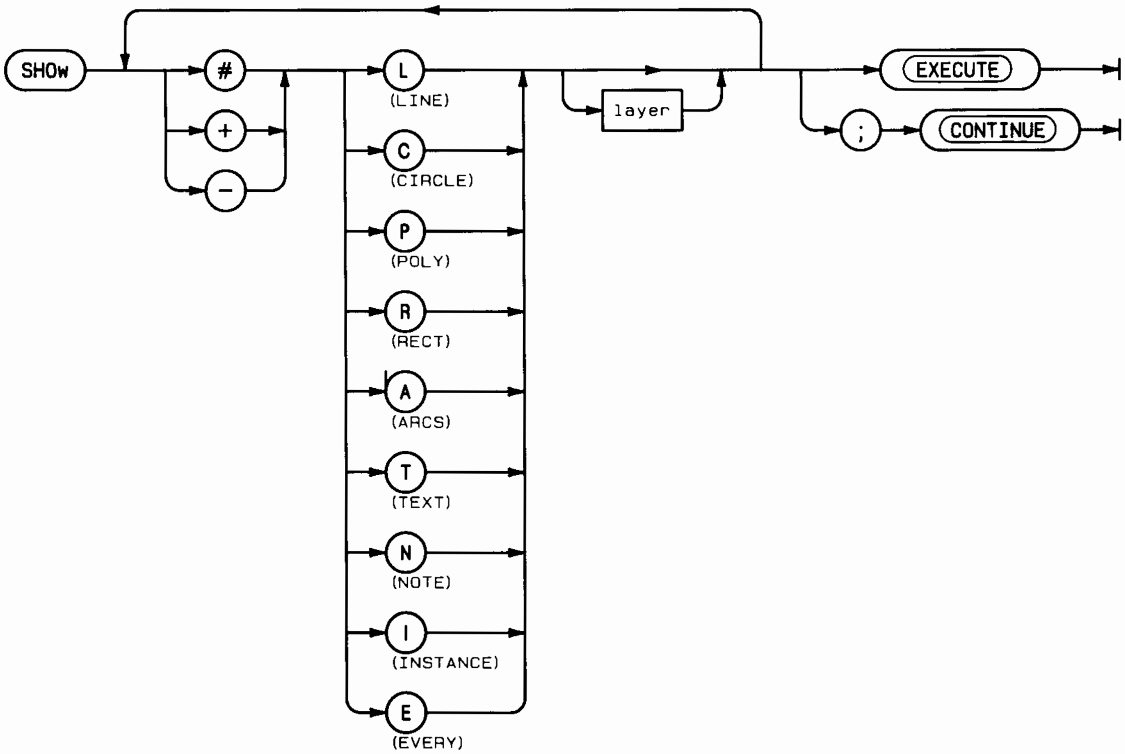
You cannot enter a macro file name that could be interpreted by the system to be a command name or a macro name. For example, entering `SAVE WIND` could cause the system to reject the SAVE command since WIND is a valid entry for the WINDOW command. Similarly, if a macro named ARCS is defined, entering `SAVE ARCS` would cause the system to reject the SAVE command.

Supplying a macro file name of an existing macro file (by directly entering the file name or by using a default file name), destroys the macro definitions in the existing file when the new macro file is created.

SHOW

Valid in:
Edit subsystem

The SHOW command allows you to specify which layers are to be displayed, which components are to be displayed in the layers, which components are modifiable, and which components may not be modified.



Item	Description/Default	Range Restrictions
#	The component(s) on the layer specified may be displayed and modified.	
+	The component(s) on the layer specified may be displayed but not modified.	
−	The component(s) on the layer specified may be neither displayed nor modified.	
A, C, E, I, L, N, P, R, T	Descriptor specifying which components are affected by the SHOW command on the layer specified.	
layer	Specifies to limit the action of the SHOW command to the layer specified. The default is all layers.	1 - 255

HELP syntax

```
SHOw {#|+|-}desc[layer] ... EOC
```

Use

The SHOW command allows you to specify which layers are active in the drawing area, and which layers are "turned off". The effects of multiple SHOW commands is accumulative.

For example, entering `SHOW -E +L3 #A1 ;` first turns off all components in all layers (components in any layer are not displayed). Then it turns on line components in layer 3 (allows them to be displayed; however, they cannot be modified). Finally, it turns on arc components in layer 1 (allows them to be both displayed and modified).

Example:

```
SHOW #E -T -R2 ;
```

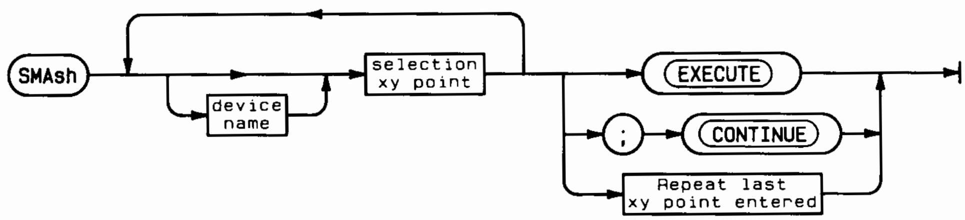
result:

> All components in all layers are turned on with the #E option. Text in all layers is turned off with the next option (–T). Finally, all rectangle components in layer 2 are turned off via –R2.

SMASH

Valid in:
Edit subsystem

The SMASH command allows you to smash a link (instance) to a device in your drawing. Smashing the link causes the components forming the linked device to be moved from nesting level 2 to nesting level 1 in the display. The link to the device (the instance component) is removed from your drawing and replaced with the components forming the device. The copy of the device in computer memory is not affected.



Item	Description/Default
device name	Limits the selection of the link to be smashed to instance components with the name specified. This valid only for the xy selection point immediately following the device name.
selection xy point	Xy point lying in or on the instance to be smashed. Entering additional selection points allows you to reselect the device to be smashed. No link is smashed until the command is terminated.

HELP syntax

```
SMAsh [devicename] xysel EOC
```

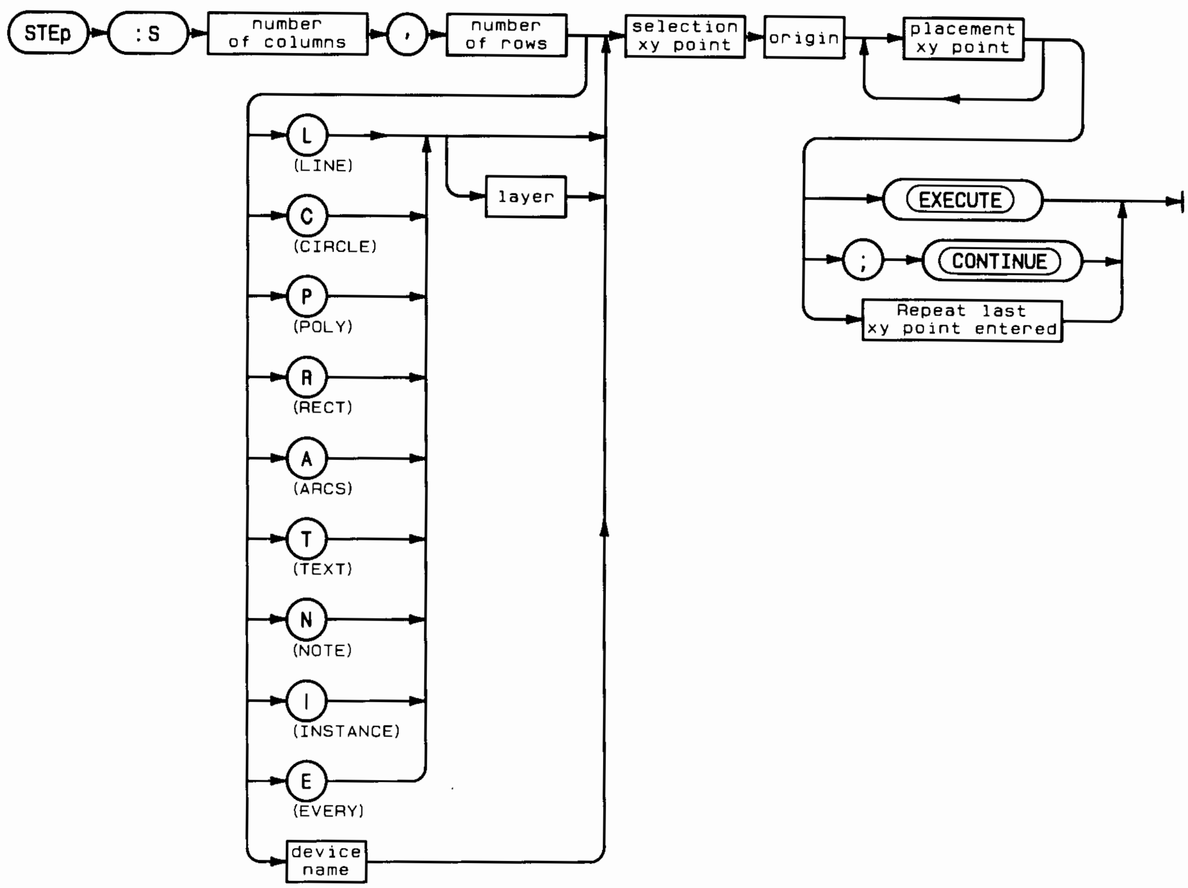
Use

Enter the SMASH command and identify the instance to be smashed by selecting an xy point that lies in or on the boundary of the instance. The device name of the instance component (link) may be used to further limit selection of the link to be smashed. Specifying additional selection points allows you to reselect the device to be smashed.

STEP :S

Valid in:
Edit subsystem

The STEP :S command allows you to create a rectangular matrix of copies of any component in the drawing area of the display. The STEP :S command is similar to the COPY command except that many copies are created, not just one.



Item	Description/Default	Range Restrictions
number of columns	Number of columns in the matrix of copies of the selected component.	
number of rows	Number of rows in the matrix of copies of the selected component.	
A, C, E, I, L, N, P, R, T	Descriptor identifying the type of component to be selected and copied. The default descriptor is, E.	
layer	Layer containing the component to be copied.	1 - 255
device name	Name of a specific instance component to be copied.	
selection xy point	Xy location lying on or inside the component you wish to copy.	
origin	Xy location which acts as a reference point for copying.	
placement xy point	This placement point specifies the X and Y distances between the reference points of the components in the matrix (see the drawing below).	

HELP syntax

```
STEp :Snx,ny [(desc[layer]|devicename)] xysel xyref xy1 [xy2...] EOC
```

Use

Entering the STEP :S command allows you to make many copies of a single component. The component to be copied is identified by entering an xy point that lies on or inside the component. You may limit selection of the component to be copied by entering an optional descriptor an layer number. If the component to be copied is an instance component, you may limit selection to this component by entering its name.

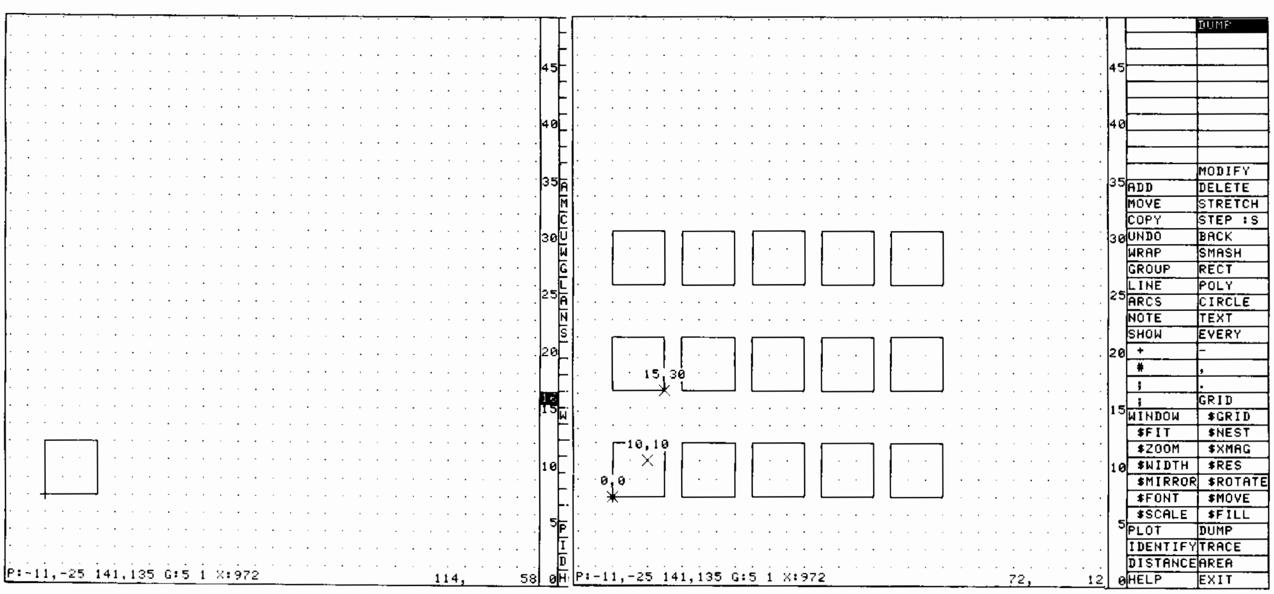
An origin must be specified for the component to be copied so that the copies may be accurately placed in the matrix. The placement xy point defines the distances between the origin of each component in the matrix (see the drawing below). Entering an additional xy placement point causes the old placement point to be replaced with the new placement point, thus redefining the distance between components in the matrix.

Example:

```
STEP :S5,3 R  10,10  0,0  20,30 ;
```

result:

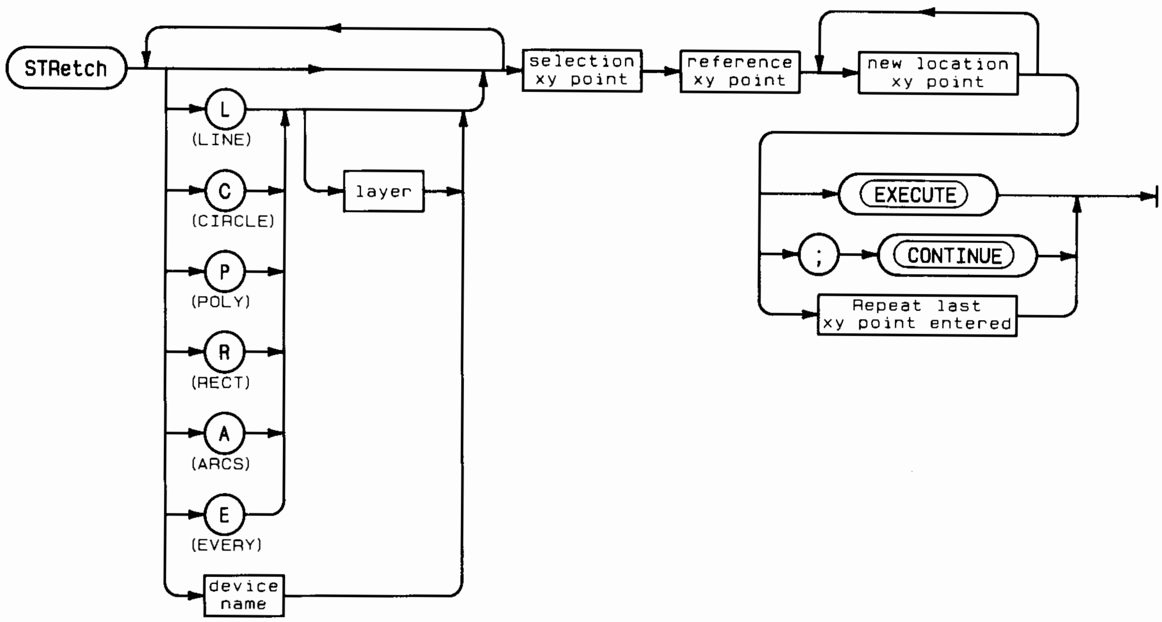
Fifteen copies of a rectangle located at 10,10 are created. These copies are arranged in a 5 (column) by 3 (row) matrix. An origin of 0,0 is assigned to the original rectangle and the copies in the matrix are spaced such that their origins are separated by 20 units in the X direction and 30 units in the Y direction.



STRETCH

Valid in:
Edit subsystem

The STRETCH command allows you to modify a component by stretching a side, edge, or vertex of the component. Text, note, and instance components cannot be stretched.



Item	Description/Default	Range Restrictions
A,C,E,L,P,R	Limits selection of the component to be stretched to the component type specified (such as R for rectangle).	
device name	Name of an instance. Since instance components cannot be stretched, this option provides no usable features.	
layer	Layer containing the component to be stretched. Only the following components can be stretched: lines, arcs, circles, polygons, and rectangles.	1 - 255
selection xy point	Point lying in or on the component to be stretched.	
reference point	Xy point identifying the part of the component to be stretched.	
new location point	Xy point identifying the location that you want to move the reference point. Entering additional xy points replaces the new location point with the new point(s) entered, thus acting as a trial placement location.	

HELP syntax

```
STRetch [{desc[layer]|devicename}] xysel xyref xy1 [xy2...] EOC
```

Use

Components are modified by selecting a component, identifying the portion of the component that you wish to modify (vertex, edge, etc), and finally specifying the new location of that portion of the component.

When a circle component is stretched, only the radius may be altered.

When an arc is stretched, you may move either endpoint of the arc or you may change the radius of the arc.

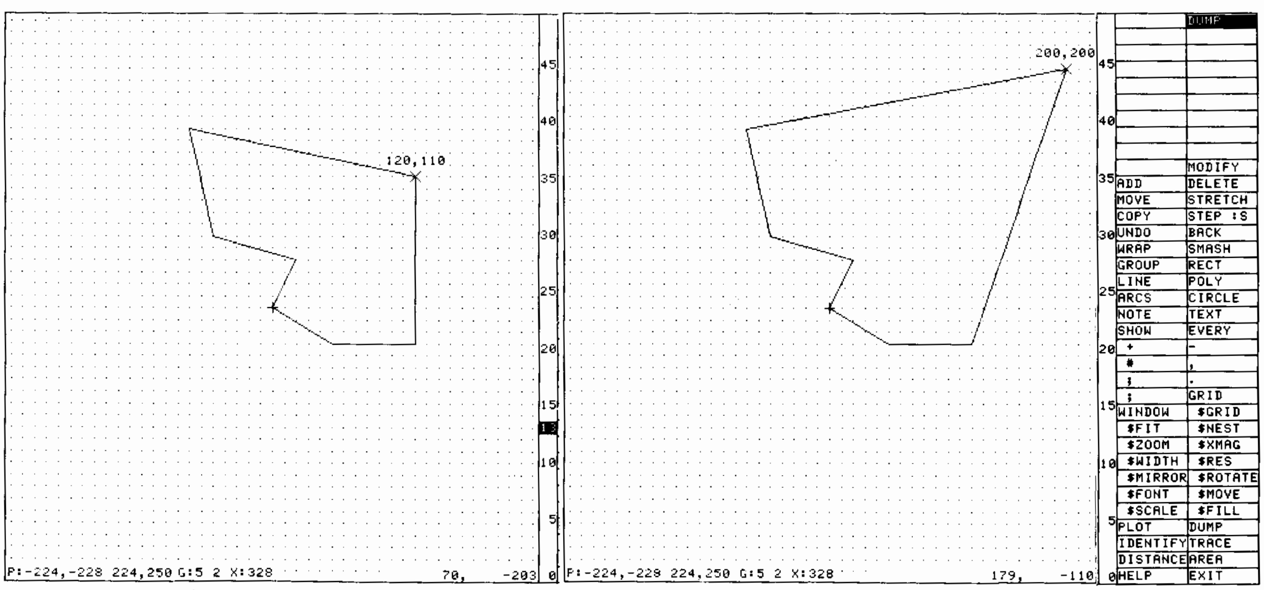
When lines, rectangles, and polygons are stretched, it is possible to move any edge, vertex, or endpoint of the component.

Text, notes, and instance components cannot be stretched.

Example:

```
STRETCH 100,100 120,110  200,200 ;
```

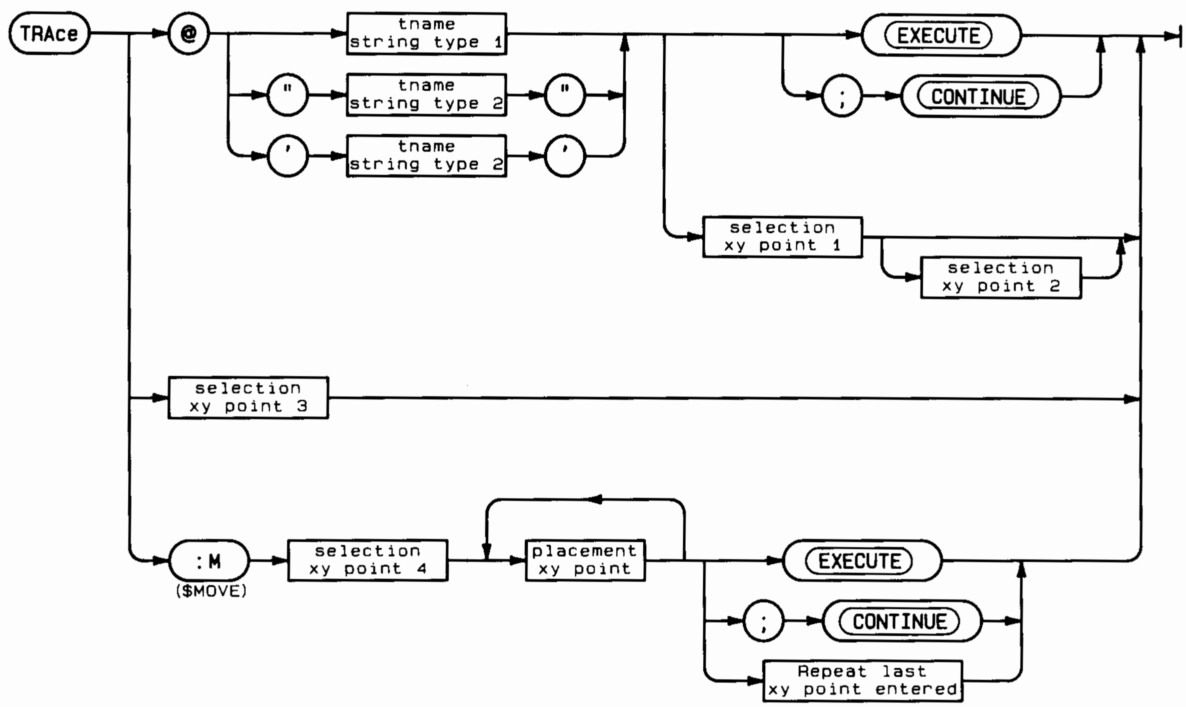
result:



TRACE

Valid in:
Edit subsystem

The TRACE command allows you to: display the tname of a particular component, change the tname of a particular component, highlight all components with the same tname, and move the location of a tname relative to the component with which it is associated.



Item	Description/Default	Range Restrictions
tname string type 1	Set of characters forming the tname that you wish to display or to assign. All characters may be displayed and/or plotted.	legal characters: A-Z, 0-9, (_), (*), ($), (?), (=)
tname string type 2	Set of characters forming the tname that you wish to display or to assign. All characters may be displayed, however only those characters common to string type 1 and string type 2 may be plotted.	legal characters: A-Z, 0-9, (<), (>), (/), (\), (%), (!), ( ), (,), (.), (;), (+), (−), (#), (*), (∧), (@), ([), (]), (_), ($), (?), (=)
selection xy point 1	Xy point lying in or on the component. Entering xy point 1 causes the tname of the component to be displayed, if the component has a tname. If the component does not already have a tname, the tname specified with TRACE command is assigned to the component.	
selection xy point 2	Xy point lying in or on the component. Entering this xy point assigns the tname entered with the TRACE command to as the new tname of the component selected with xy point 1.	

selection xy point 3	Xy point lying in or on a component. Entering this point causes the tname assigned to the component to be displayed.
selection xy point 4	Xy point lying in or on a component. This point identifies the component whose tname you wish to move.
placement xy point	Xy point specifying the new location of the tname. The tname is moved such that the lower left corner of the tname corresponds to the placement xy point. Entering additional placement points allows you to replace the previous placement point with the new placement point, thus the placement point is a trial placement location.

HELP syntax

```
TRAce {xysel ... | @tname {xy1[xy2]|EOC} | :M xysel [xypnt ...] EOC}
```

Use

The TRACE command allows you to: display the tname of a particular component, change the tname of a particular component, highlight all components with a particular tname, and move the location of an tname relative to the component with which it is associated.

Entering TRACE with a tname string allows you to assign the tname to the component selected. If the component already has a tname, the first xy selection point causes it to be displayed while the second xy selection point causes the old tname to be replaced with the new tname. If the component does not have a tname, the tname entered with the TRACE command is assigned to the component when the first xy selection point is entered.

Entering the TRACE command with only a selection point causes the tname of the component selected to be displayed.

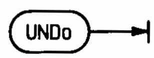
Entering the TRACE command with the :M option allows you to move the location of the tname assigned to a component. To move the tname, select a point that lies in or on the component whose tname you wish to move. Then select the xy location where you wish the tname to be plotted on the display.

When a component is selected with the TRACE command, the component type and the layer number on which it is entered are displayed in the upper left corner of the CRT.

UNDO

Valid in:
Edit subsystem

The UNDO command reverses the effects of the command immediately preceding it. The UNDO command allows you to reverse the effects of the following commands: ADD, COPY, DELETE, GROUP, MODIFY, MOVE, SMASH, STEP :S, STRETCH, and WRAP.



HELP syntax

```
UNDo <undoes ADD,COPy,DELete,GROup,MODify,MOVe,SMAsh,STEp,STRetch,WRAp>
```

Use

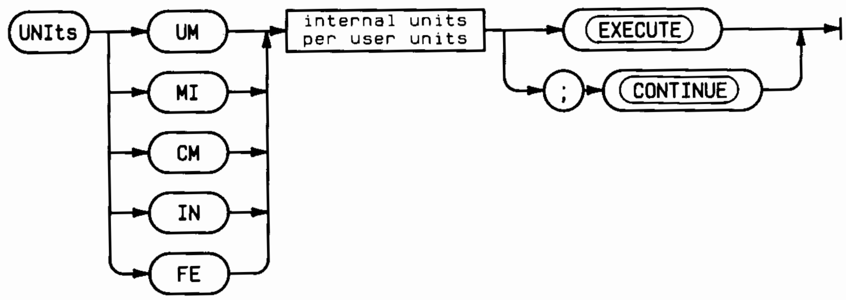
The UNDO command reverses the effects of the last command entered. For example, if the previous command was to delete a rectangle, entering the UNDO command would restore the rectangle to the drawing.

The UNDO command allows you to reverse the effects of the following commands: ADD, COPY, DELETE, GROUP, MODIFY, MOVE, SMASH, STEP :S, STRETCH, and WRAP.

UNITS

Valid in:
Process subsystem

The UNITS command allows you to assign a physical distance to the distance between system grid points, thus it determines the units of your drawing.



Item	Description/Default	Range Restrictions
UM	Specifies micrometers as the units.	
CM	Specifies centimeters as the units.	
MI	Specifies mils (.001 inches) as the units.	
IN	Specifies inches as the units.	
FT or FE	Specifies feet as the units.	
system grid points per user unit	Specifies the number of system grid points that are equal to one user unit. For example, `UNITS CM, 2` specifies that the distance between two (2) system grid points is equal to one centimeter.	1 - 16383

HELP syntax

```
UNIts {UM | CM | MIl | INch | {FT | FEet} } resolution
```

Use

Enter the UNITS command to specify the units of the drawing and to determine the resolution that may be accessed in the drawing. For example, suppose that the units are defined by `UNITS IN, 100`. This means that the distance between 100 system grid points is equal to 1 inch. Since an xy location can only be entered on a grid point, the resolution to which you can specify locations is limited to .01 inches. In other words, the distance between each system grid point represents .01 inch.

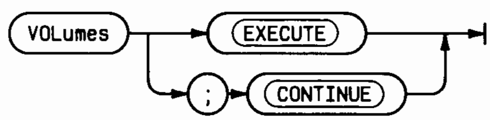
If a device is present in computer memory, changing the user units causes the system to issue a message that the new user units do not match the units of the device(s). You are then asked if you still wish to change the user units. Responding N for no causes the current user units not to be changed. Entering Y for yes causes the system to automatically convert the units of the drawing to match the new user units.

For example, suppose that the units of the device are one inch per system grid point (`UNITS IN,1`). If the new user units are 1 foot per system grid point, the system converts the units of the device to one foot per system grid point. Thus, if a component in the device was originally 1 inch long, it is 1 foot long when displayed with the new user units.

VOLUMES

Valid in:
Graphics Editor
Edit subsystem
Macro Subsystem
Process Subsystem

The VOLUMES command displays a list of all volumes which are currently on line and recognized by the system.



HELP syntax

```
VOLumes EOC
```

Use

This command allows the user to display the volume number and volume name of all EW volumes which are currently on line. In addition, the current prefix volume will be displayed. To obtain a printed copy of this information, simultaneously press CTRL k0.

Example:

```
VOLumes;
```

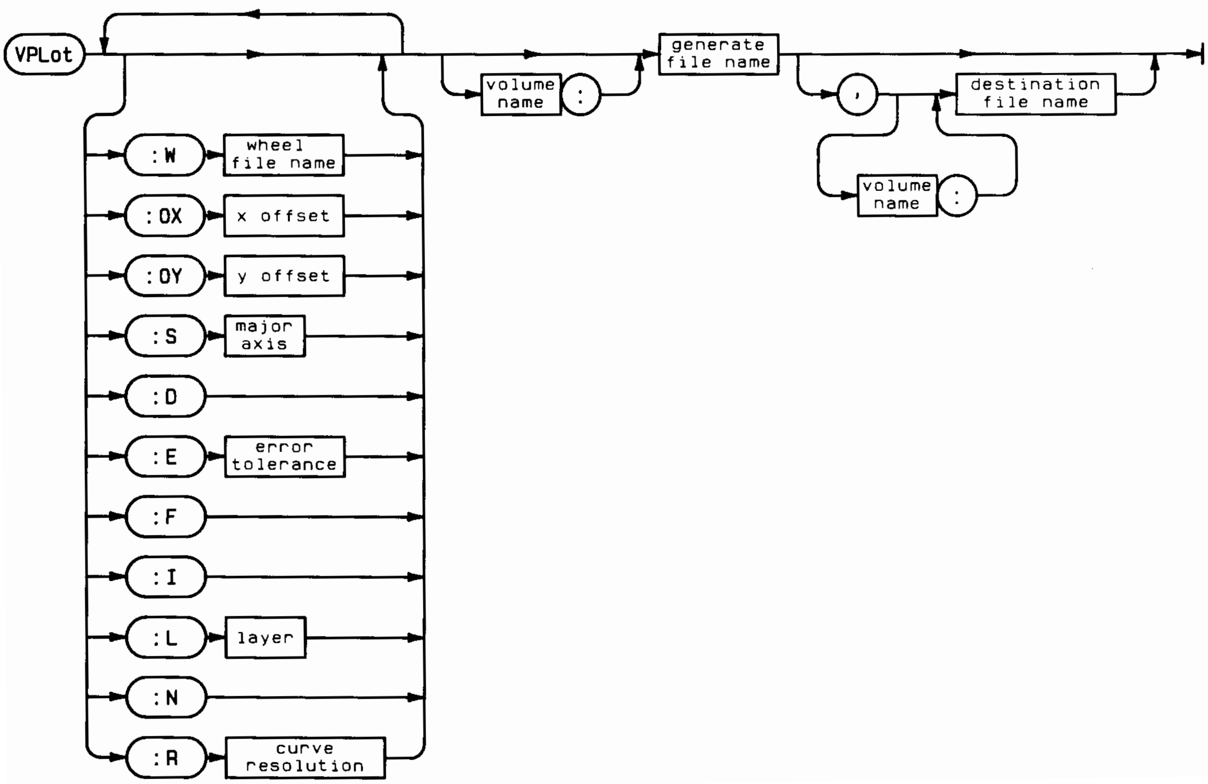
result:

```
Volumes on-line:
 3 # EWSYS:
 4 # EWPC:
 5 # EWSCH:
 7 # EWLSC:

Prefix is - EWSYS:
```

VPLOT

Valid in:
P. C. Layout module

The VPLot command causes the system to operate on one or more generate files to produce preliminary photo plot information. The system places this preliminary information in a special EW file, called a virtual plot file.



Item	Description/Default
wheel file name	Name of the wheel file which describes the aperture wheel that is to be used on the photo plotter. The wheel file must be constructed and named as detailed in the conventions section of this chapter.
x offset	The :OX parameter and the accompanying data help you center your finished photo plot on the film. VPLOT attempts to plot your drawing in the lower left hand corner of the film. The :OX parameter allows you to shift your plot on the film by the amount specified by x offset (in mils) in the positive X direction. For example, specifying :OX100 tells the system to shift the plot on the film 100 mils in the positive X direction, thus providing a 100 mil 'margin' on left edge of the film.
y offset	The :OY parameter and the accompanying data help you orient your finished photo plot on the film. Y offset tells the system to place the finished plot that number of mils in the positive Y direction from the lower left hand corner of the film. For example, specifying :OY25 specifies to plot the drawing 25 mils in the positive Y direction from the lower left hand corner of the film.
major axis	If most components are aligned along the X-axis, use 'x'. If most components are aligned along the Y-axis, use 'y'. For example, :S sorts the data optimizing along the X-axis. If neither axis predominates, use either 'x' or 'y'.
curve resolution	The :R parameter allows you to force the resolution of the photo plotted circles and arcs to something other than the values in the GENERATE file. The resolution value must be between 1° and 120°, inclusive.
error tolerance	The :E parameter and the accompanying data specifies the tolerance in the size of the component that is acceptable when photo plotting your drawing. VPLOT uses this tolerance when selecting the aperture that is used to plot a component. For example, to photo plot a 24 mil diameter circle with :E1 specified, allows the program to specify a 23 to 25 mil aperture to plot with.
layer	The :L parameter and the accompanying data tell VPLOT which logical layers of the drawing are to be photo plotted. For example, specifying :L1 :L2 :L5 with the VPLOT command causes logical layers one, two, and five of the drawing specified (and only those logical layers) to be photo plotted. Specifying, :LA, tells VPLOT to photo plot all logical layers in the drawing.
volume name	Name of the volume containing the generate file. The default volume is the volume containing the system software.
generate file	Name of the generate file containing the drawing you wish to photo plot.
volume name	Name of the volume to contain the virtual plot file. The default volume is the volume containing the system software.
destination file name	Name you wish assigned to the virtual plot file created. If no destination file name is supplied, the system uses the file name of the generate file as the virtual plot file name. The suffix .VPL is appended to the file name to allow the system to recognize the file as a virtual plot file.

HELP Syntax:

```
VPLot [:Wxxxxxxx] [:OXnnn] [:OYnnn] [:Sx] [:D] [:Erange]
[:Rres] [:F] [:I] [:L1] ...  [L1] [:N] source_file [,dest_file]
```

Use

The VPLOT command causes the system to operate on one or more generate files to produce preliminary photo plot information. The system places this preliminary information in a special EW file, called a virtual plot file. The virtual plot file is then operated on by the PPLOT program to produce the final set of commands that drives a Gerber photo plotter.

Several parameters may be specified with the VPLot command allowing many options in the final photo plot. These parameters are discussed in detail along with their effects on the photo plot in the sections below.

The :S parameter specifies that you wish VPLOT to sort the components to be plotted. Sorting the components causes the system to arrange the components to be plotted so that the photo plotter makes a minimum number of aperture changes. There is a significant time trade off involved in telling the system to sort. Sorting causes the amount of time required to run VPLOT to increase while it decreases the amount of time required to perform the actual photo plot.

The :D parameter specifies that you wish VPLOT to eliminate any duplicate components it finds. For example, if two circles have the same radius, the same resolution, the same location, and are drawn in the same logical layer, specifying :D with VPLOT causes one of the circles to be eliminated. Specifying the :D parameter forces the system to sort the components to be plotted and thus, the :S parameter is automatically set.

The :F parameter specifies to fill all components that are plotted (only circle, rectangle, and line components are 'fillable'). If the :F parameter is not entered with VPLOT, the system plots only the outlines of the components (except for those with inames that begin with the letter 'Z', which are flashed). If :F is specified, VPLOT attempts to match the dimensions of as many of the components as it can with dimensions of specific apertures on the Gerber plotter. Those which it can match, it generates commands to the plotter to flash. Those which it cannot match, it generates commands to the plotter to trace and fill.

Any component with an iname of the form Zxxxxx causes the system to check the Gerber Aperture Table for an aperture with an aperture name xxxxx. If the aperture name is found, VPLOT flashes that aperture instead of tracing and filling the component. If the aperture name is not found, the system displays a message to that effect and plots no component. Any component with the iname ZNULL is not photo plotted at all.

VPLOT is capable of filling all components except: text, notes, and polygons. Text and note components are photo plotted using the smallest circular trace aperture on the aperture wheel. Text components are photo plotted with no width and thus, resemble notes on the finished plot. Polygon components are simply outlined on the photo plot using the smallest circular trace aperture available wheel. Rectangles that have been rotated at a non-integer multiple of 90 degrees, are considered polygons by the Graphics Editor, and thus by VPLOT.

The :I parameter specifies that you wish a report printed on the system printer detailing the apertures used, the number of each type of component plotted, which parameters were specified with VPLOT, etc. If :I is not specified, no report is generated. A sample report is provided below, demonstrating the different information available.

```
    Apertures used for file EWLSC:EXAMPLE.GEN

        Aperture                               Size      Uses
ZW12              Trace Circle                 12         1
ZW20              Trace Circle                 20         2
ZW25              Trace Circle                 25         8
ZW80              Trace Circle                 80        23
ZTARGET           Flash Square                  0         2

Wheel file used: EWPC:WHEEL1


Input File: EWLSC:EXAMPLE.GEN
Output File: EWLSC:EX.SLDR

Options set: Sort (Y)  Info  Fill


Layers Plotted:
19 21 28

Error range:    plus/minus 0
X offset used:             3000
Y offset used:             3000
Origin:                  -3670,1825
Rectangles flashed:      2
Rectangles filled:       2
Total rectangles:        4
Circles flashed:         0
Circles filled:          3
Total circles:           3
Lines traced:            28
Lines filled:            0
Total lines:             28
Polygons outlined:       0
Arcs traced:             0
Characters of text:      13
Special names flashed:   2
Null names thrown away:  10
Draws:                     157.0
Moves:                      56.0
Flashes:                 2
Aperture changes:        7

Total components output: 36
```

The :N parameter specifies that VPLOT echo data on the system CRT as it is read from the wheel file and the individual generate files.

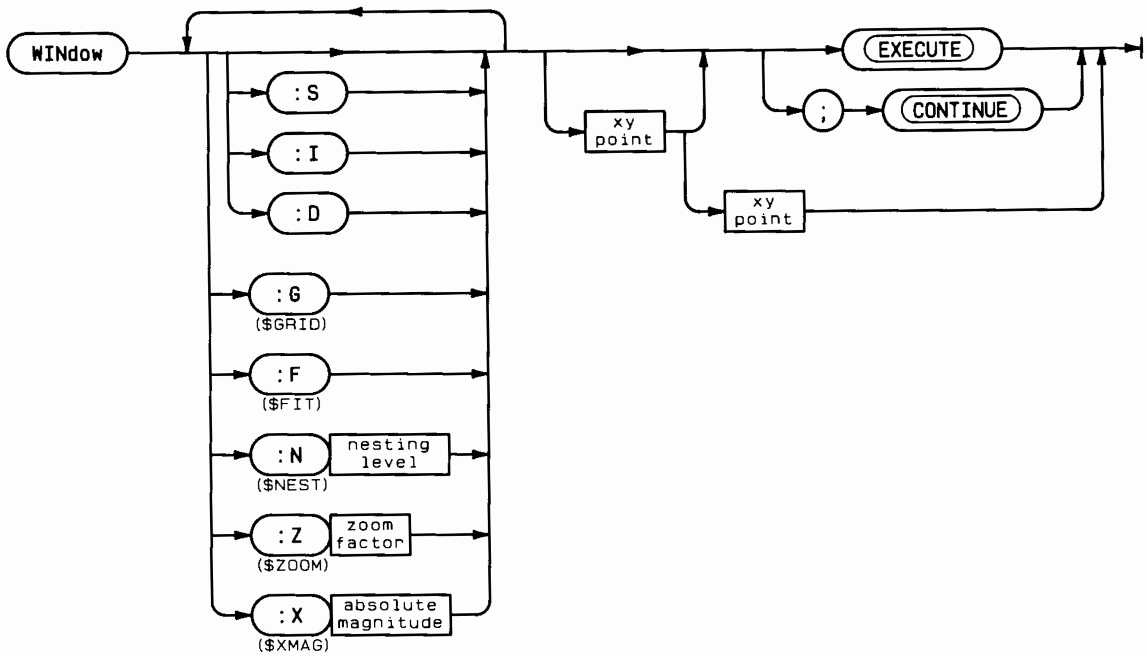
Once the VPLOT command has been entered, the system reads and processes the generate file and then prompts you for additional generate files. Entering additional file specifiers causes the system to read the additional generate files and combine the information in the final photo plot. Entering no file specifier and pressing the CONTINUE key causes the system to assume that you are finished inputting file specifiers. The system then creates the virtual plot file.

WINDOW

Valid in:

Edit subsystem

The WINDOW command allows you to: control the display mode, control the depth of nesting level displayed, control the grid display, redraw the screen, and control the magnification and the location of the window over the drawing.



Item	Description/Default	Range Restrictions
:S	Sets the display in the symbolic mode. The default mode of the display is the detail mode.	
:I	Sets the display in the interconnect mode. The default mode of the display is the detail mode.	
:D	Sets the display in the detail mode. The default mode of the display is the detail mode.	
:G	Specifies to toggle the grid on or off when the screen is redrawn. If the grid was displayed, it will be turned off when the screen is redrawn. Similarly, if the grid was off, it will be displayed when the screen is redrawn.	
:F	Specifies to redraw the screen and adjust the window such that the entire drawing, fits into the drawing area of the CRT.	
nesting level	Specifies the depth of nesting levels to be displayed in the drawing area. Entering :N4 specifies to display components of devices in nesting levels 1, 2, 3, and 4. The nesting depth at entry to the edit subsystem is nesting level 1.	

zoom factor	Specifies the magnification of the drawing displayed on the CRT. Positive values (>0) cause current window parameters to be multiplied by the zoom factor. Negative values (<0) cause the window parameters to be divided by the zoom factor. Since the window parameters control the size of the drawing, the net effect is to cause the display to zoom in or zoom out on the drawing. If no zoom factor is supplied with the :Z parameter, the window is adjust to match the window setting previous to the one you are viewing. For example, if you change the window setting from A to B (via a WINDOW command), entering WINDOW :Z ; changes the window setting back to A. Entering WINDOW :Z ; again changes the window setting back to B. Thus, this feature allows you to toggle between 2 different window settings by simply entering WINDOW :Z ;.	
absolute magnification	Specifies the absolute magnification of the drawing. This specifies the relationship between user units and the magnification of the drawing in the display. For example, if the user units are inches, specifying :X2 causes a device 1 user unit long (1'' long) to occupy 2 inches of the CRT when displayed.	
xy point 1	If xy point 1 alone is entered (i.e., xy point 2 is not entered) the window is shifted such that the xy point 1 becomes the center of the drawing. If xy point 2 is entered with xy point 1, xy point 1 specifies one corner of the new window.	
xy point 2	Specifies the second corner of the new window. The window is changed such that the smallest x value and the smallest y value of the two points becomes the location of the lower left corner of the window. The magnification of the drawing viewed through the window is then changed such that the entire rectangular area defined by xy points 1 and 2 is contained in the window.	1 - 500

HELP syntax

```
WINdow [<:S|:I|:D>] [:G] [:F] [:Nlevel] [:Z[pwr]] [:Xmag]
[xy1 [xy2]] EOC
```

Use

Entering the WINDOW command allows you to set the mode of the display.

- In the interconnect mode, only components entered on layers defined as interconnect layers are displayed.
- In the symbolic mode, all components entered on layers defined as symbolic layers and interconnect layers are displayed. Lines with width in interconnect layers are displayed with 0 width.
- In the detail mode, all components entered on layers defined as detail layers and as interconnect layers are displayed.

Entering just the WINDOW command, with no parameters, causes the screen to be redrawn.

Several options are provided for determining the size and location of the window. If conflicting options are entered (such as :Z2 and :F or :S and :D) the system selects the last one entered as the option to be used.

Example:

```
WINDOW  :S :N3 :Z5 ;
```

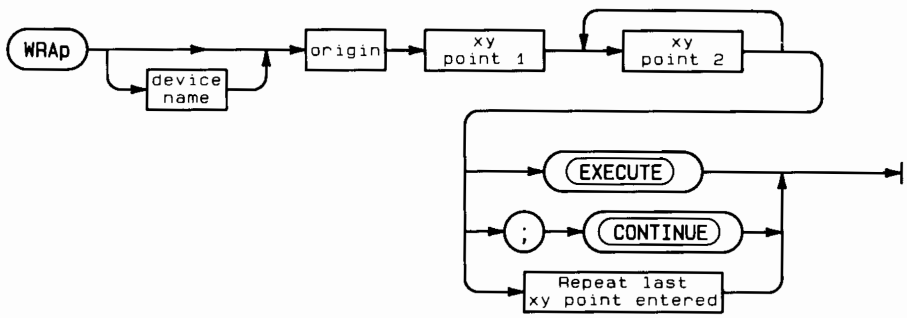
result:

The display is set in the symbolic mode and the nesting depth displayed is set to 3 (components in nesting levels 1, 2, and 3 are displayed). Additionally, the display zooms out by a factor of 5.

WRAP

Valid in:
Edit subsystem

The WRAP command allows you to create a device from components in the drawing area of the display. A copy of the new device is placed in computer memory. The components forming the device are displayed in nesting level 2. An instance component replaces the components in nesting level 1, linking the new device to your drawing. No more than 500 devices may be wrapped at one time.



Item	Description/Default	Range Restrictions
device name	Name assigned to the device created.	8 characters max. legal characters: A-Z, 0-9, (_). The first character of name cannot be a number.
origin	Xy location specifying the origin of the device that you create.	
xy point 1	Specifies the location of one corner of the wrapping rectangle.	
xy point 2	Specifies the location of the corner of the wrapping rectangle that lies on the diagonal from xy point 1. Xy point 2 is a trial placement location. Entering additional xy points allows the previous xy point 2 to be replaced with the new xy point 2, thus allowing you to change the size of the wrapping rectangle.	

HELP syntax

```
WRAp [devicename] xyorg xy1 xy2 [xy2...] EOC
```

Use

The WRAP command allows you to create a device from selected components on the CRT. Two data points specified with the command define opposite corners of a wrapping rectangle. Only components **completely** enclosed by the wrapping rectangle become part of the new device.

Entering a name with the WRAP command causes a library part to be created from the device when the drawing is saved. If the name entered already exists as a device name, the system will not allow you to use that name and the command must be re-entered. Additionally, you cannot enter a name for the device that could be interpreted by the system to be a command name or a macro name. For example, entering `WRAP WIND` ... would cause the system to reject the WRAP command since WIND is a valid entry for the WINDOW command. Similarly, if a macro named ARCS is defined, entering `WRAP ARCS` ... would cause the system to reject the WRAP command.

If no device name is entered, the device created cannot be stored and cannot be added to the drawing (no library part is created when the drawing is saved with the SAVE command). However, an instance of the device can still be copied, moved, smashed, etc, just like any other component. The device is active until you exit the Graphics Editor, and then is lost.

The origin specified with the command is used to accurately place the instance component. For example, after a device is created, adding an instance of the device to a drawing at the location 10,10 causes the instance to be added such that its origin is placed exactly at the location 10,10.

Example:

```
WRAP TEST  0,0   25,10  200,100  ;
```


result:

> A new device named TEST is created with an origin located at 0,0. The device is composed of all components with in the wrapping rectangle defined by the locations 25,10 and 200,100. If the drawing in which the device is created is saved with the SAVE command before you exit the Graphics Editor, the device becomes a library part (is stored in a device file).
>
> If the drawing in which the device is created in not saved, you may still save the device in a device file by first exiting the edit subsystem. Next edit the device that you wish to save (enter the edit subsystem by entering EDIT devicename ;). Now the device can be saved by entering the SAVE command.

Appendix A

EGS-45 operates on the HP 9845 computer equipped with an English, French, German, Spanish or Swedish-Finnish keyboard. When a non-English keyboard is used, typing certain keys causes their English equivalent to be displayed.* Other keys are completely ignored. The tables below show which keys are affected.

French Keyboard

Typing this key:	Displays this character:
à	a
ç	c
è	e
é	e
ù	u
ˆ	(ignored)**
¨	(ignored)
°	(ignored)

German Keyboard

Typing this key:	Displays this character:
ä	a
Ä	A
ö	o
Ö	O
ü	u
Ü	U
B	(ignored)

Spanish Keyboard

Typing this key:	Displays this character:
ñ	n
Ñ	N
°	(ignored)
´	(ignored)
¨	(ignored)
¡	(ignored)
¿	(ignored)

Swedish-Finnish Keyboard

Typing this key:	Displays this character:
å	a
Å	A
ä	a
Ä	A
ö	o
Ö	O

* EW environment only. See Chapters 2 and 3 for information on EW environment.

**The accent mark (ˆ) is ignored. The caret (∧) located in the numeric keyboard is still functional and is used in the Edit Subsystem.

Subject Index

a

b

c

d

e

f

g

h

i

l

m

n

o

p

r

S

t

u

v

w

z

HEWLETT PACKARD

Part No. 98301-12111
E0782

Printed in U.S.A.
First Edition, July 1982