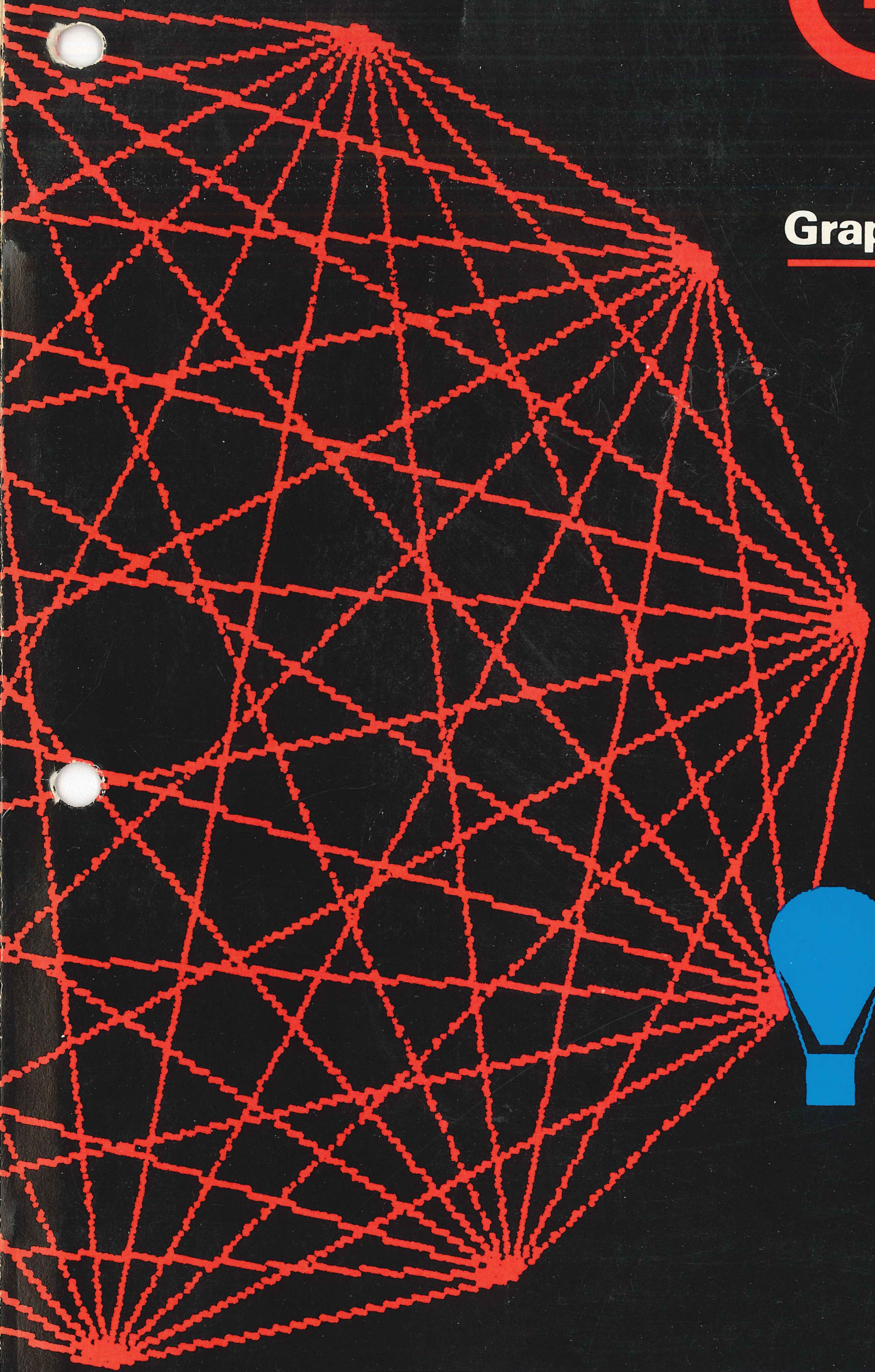
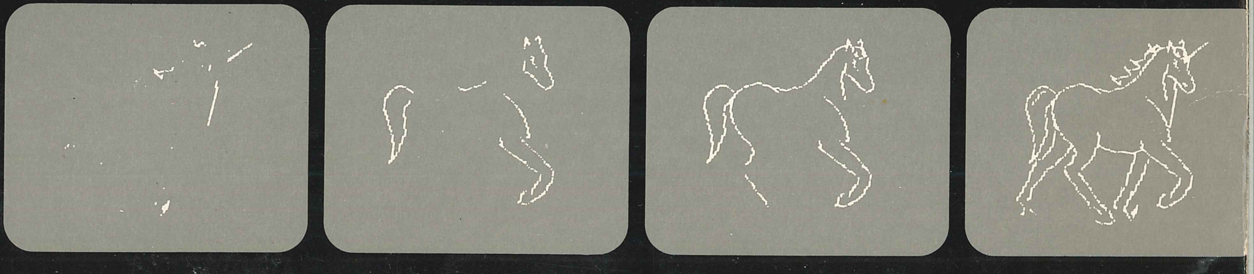
GIGI™

Graphics Editor Manual





digital

DIGITAL EQUIPMENT CORPORATION

February 1981

GIGI™ Graphics Editor Manual

Order No. AA-J942A-TK

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

DIGITAL EQUIPMENT CORPORATION

First Printing, February 1981

DEC	FOCAL
DECnet	IAS
DECsystem-10	PDP
DECtape	RSX
DECUS	UNIBUS
DIBOL	VAX
DIGITAL	VMS

CONTENTS

PREFACE

GIGI (Graphics Image Generator and Interpreter) is a micro-processor based keyboard that functions as a graphics terminalwhen connected to either a color or a black and white monitor. The GIGI Graphics Editor is supported by three operating systems: VAX/VMS, TOPS-20, and RSTS/E.

GIGI includes several firmware and software components to complement its hardware facilities. A set of manuals describes the use of these components and their interfaces. These manuals are:

DOCUMENTATION

GIGI BASIC Manual
(AA-K335A-TK)

GIGI/ReGIS Handbook
(AA-K336A-TK)

GIGI Character Set Editor Manual
(AA-K337A-TK)

GIGI Slide Projection System Manual
(AA-J943A-TK)

GIGI COMPUTER ASSISTED INSTRUCTION MATERIALS

VAX/VMS GIGI/ReGIS CAI Primers
(BE-K391A-BC TU58)
(AS-K327A-BE Floppy)

RSTS/E GIGI/REGIS/ CAI
(BC-K346A-BC RL02)
(AP-K392A-BC Magtape 9 track 800 BPI)
(BB-K393A-BC Magtape 9 track 1600 BPI)

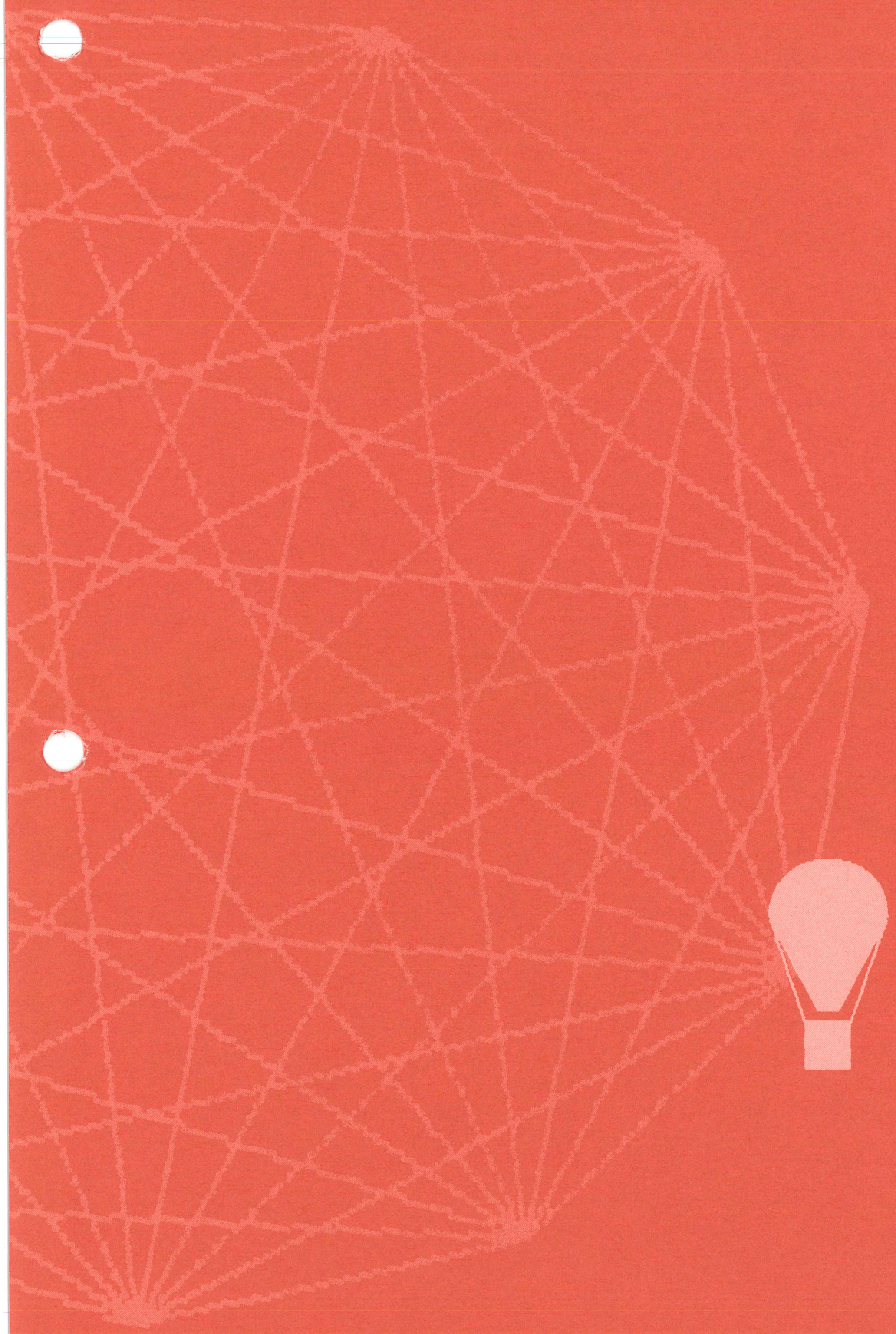
GIGI/ReGIS CAI Primers
(AA-K329A-TE)

VAX/VMS GIGI/REGIS CAI Primers Course Administrator's Guide
(AA-K328A-TE)

RSTS/E GIGI/REGIS CAI Primers Course Administrator's Guide
(AA-K347A-TC)

Before using the Graphics Editor, you should read the GIGI ReGIS Handbook and the GIGI Primer to familiarize yourself with the use of GIGI.

1 Introduction to GIGI



1 INTRODUCTION TO GIGI

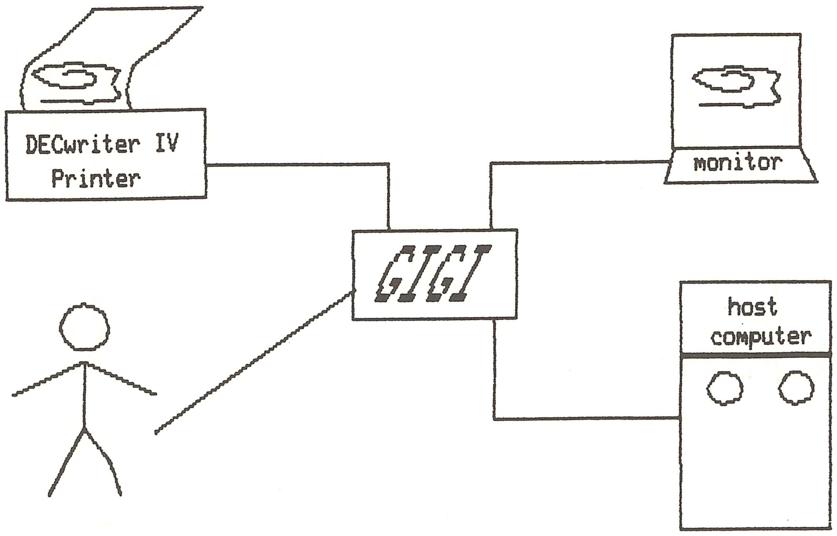
The GIGI Graphics Editor is a software tool designed to allow you to easily use the graphics capabilities of GIGI. With the Graphics Editor you can create and modify pictures, read and write files containing picture descriptions, and display pictures.

GIGI AND THE GRAPHICS EDITOR

The Graphics Editor is a software package that utilizes the graphics capabilities of GIGI, the Graphics Image Generator and Interpreter terminal. The Graphics Editor can run under any of three operating systems:

- VAX/VMS
- TOPS-20
- RSTS/E

GIGI and a monitor are necessary to use the Graphics Editor. The Graphics Editor supports both color and black and white monitors. For hardcopy displays, the Graphics Editor supports the DECwriter IV Graphics Printer, the LA34. The diagram below represents a GIGI configuration under which the Graphics Editor runs.



Graphics Editor Capabilities

The Graphics Editor provides programmed keys and typed commands to use its many features. These features include:

- functions with which you can draw objects
- functions that allow you to manipulate screen objects
- attributes with which you can enhance your drawings: colors, blinking, patterns, shading
- use of character sets created with the Character Set Editor or ReGIS
- text modification attributes, including spacing, italics, height, width

The graphics in this manual have been created using the Graphics Editor and the DECwriter IV Graphics Printer. These provide an indication of the capabilities of the Graphics Editor. Using a color monitor allows you to use the full capabilities of the Graphics Editor, including such features as blinking, color, and the creation of animated characters.

OTHER GIGI SOFTWARE

GIGI supports other software that is compatible with the Graphics Editor. These packages are described briefly below.

Character Set Editor

The Character Set Editor is an interactive software package used to create character sets. The Character Set Editor can be used in applications where Greek or APL fonts are necessary. It can also be used to create special figures, such as arrows, molecules, and math symbols.

Character sets created with the Character Set Editor can be loaded into the GIGI and accessed and displayed with the Graphics Editor. For more information about the Character Set Editor, refer to the Character Set Editor Manual.

Slide Projection System

The Slide Projection System is an interactive software package used to display sequences of pictures created with the Graphics Editor. With the Slide Projection System, you can arrange series of pictures into sequences and display these in a manner similar to using a film slide projector. For more information, refer to the Slide Projection System Manual.

INVOKING THE GRAPHICS EDITOR

The GIGI Graphics Editor is available for use on three operating systems:

- VMS on VAX systems
- RSTS/E on PDP-11 systems
- TOPS-20 on TOPS-20 systems

Each of these operating systems displays a prompt on the command line and unique methods of logging into the system. The following paragraphs list manuals containing information on how to log on and describe how to invoke the Graphics Editor on each of these systems.

These sections below show command lines that may be modified by your system manager. If you have problems invoking the Graphics Editor on your system, ask your system manager for help.

Invoking the Graphics Editor on VAX

The document VAX/VMS Primer, order number AA-D030A-TE, provides information on how to log on to the VMS operating system and get started using the system. This document also references other information you may need to get started using VMS.

VMS prompts you with a dollar sign on the command line. To invoke the Graphics Editor on the command line, type RUN SYS$SYSTEM:GE and press the RETURN key:

```
$RUN SYS$SYSTEM:GE RETURN
```

The Graphics Editor then displays the screen and prompts you for a command.

Invoking the Graphics Editor on RSTS/E

The document RSTS/E System User's Guide, order number AA-5133B-TC, provides information on how to log on to the RSTS/E operating system and get started using the system. This document also references other information you may need to get started using RSTS/E.

RSTS/E prompts you with "Ready". To invoke the Graphics Editor, type RUN $GE and press the RETURN key:

```
Ready
RUN $GE RETURN
```

The Graphics Editor then displays the screen and prompts you for a command.

Invoking the Graphics Editor on TOPS-20

The document Getting Started with TOPS-20, order number AA-4187D-TM, provides information on how to log on to the TOPS-20 operating system and get started using the system. This document also references other information you may need to get started using TOPS-20.

TOPS-20 prompts you with an at-sign (@) on the command line; to invoke the Graphics Editor, type RUN PS:<SUBSYS>GE and press the RETURN key:

```
@RUN PS:<SUBSYS>GE RETURN
```

The Graphics Editor then displays the screen and prompts you for a command.

HOW TO USE THE GRAPHICS EDITOR MANUAL

This manual is divided into four parts.

Part I contains the introduction and presents an overview of the Graphics Editor and GIGI.

Part II is a usage section. It contains:

- **Chapter 2, Graphics Editor Concepts,** which describes the underlying concepts the Graphics Editor uses
- **Chapter 3, Graphics Editor Keyboard,** which describes the GIGI keyboard and how it is used with the Graphics Editor
- **Chapter 4, Files and Saving Pictures,** which describes the Graphics Editor's file handling capabilities
- **Chapter 5, Drawing Pictures,** which describes how to draw pictures
- **Chapter 6, Modifying Pictures,** which describes how to modify pictures

Part III is a reference section. It contains:

- **Chapter 7, Auxiliary and Main Keypad Keys - Reference,** which provides reference material for the GIGI keyboard as used by the Graphics Editor
- **Chapter 8, Commands-Reference,** which provides reference material for the typed commands

Part IV contains the glossary, which includes both GIGI and Graphics Editor terms, and appendix A, which describes the ReGIS input translator.

2 GRAPHICS EDITOR CONCEPTS

This chapter presents the underlying concepts used by the Graphics Editor. It starts by describing what pictures are from the lowest level of a single dot on the screen to the Graphics Editor functions that allow you to put shapes on the screen. This chapter also presents a brief conceptual overview of the Graphics Editor: typed and keypad commands, and graphics and command cursors.

PICTURES

A picture is the display on the screen. Pictures can include such things as circles, boxes, lines, curved lines, and text. The entire screen display is referred to as a picture. The Graphics Editor provides commands to allow you to create and save pictures.

Pixels

The smallest part of a picture is a pixel or picture element. The screen display consists of a total of 768 pixels on each horizontal line and 480 pixels on each vertical line. Of these, GIGI can reference all 768 pixels on each horizontal line and 240 pixels on each vertical line. GIGI uses these pixels as reference points to draw pictures.

ReGIS

The Graphics Editor uses ReGIS (Remote Graphics Instruction Set) to draw pictures. ReGIS contains commands that represent screen, writing, and text functions. Screen functions are functions that affect the overall appearance of the screen, such as screen color and timing parameters. Writing functions are functions that affect the objects that are drawn on the screen, such as lines, curves, and circles. Writing functions have attributes that you can define, such as blinking, line patterns, and shading. Text functions are functions that allow text to be included in the picture. Text can be either the standard ASCII character set or character sets that you can design with the Character Set Editor or directly with ReGIS.

The Graphics Editor draws pictures by displaying these points. The Graphics Editor has commands that correspond to the ReGIS functions. The Graphics Editor LINE, CURVE, CIRCLE, and BOX functions correspond to the ReGIS writing functions; the Graphics Editor TEXT function corresponds to the ReGIS text function; and the Graphics Editor Screen Color attribute and the ERASE function correspond to the ReGIS screen function.

You do not need to know ReGIS to run the Graphics Editor. For more information about ReGIS, refer to the *GIGI/ReGIS Handbook*.

Points

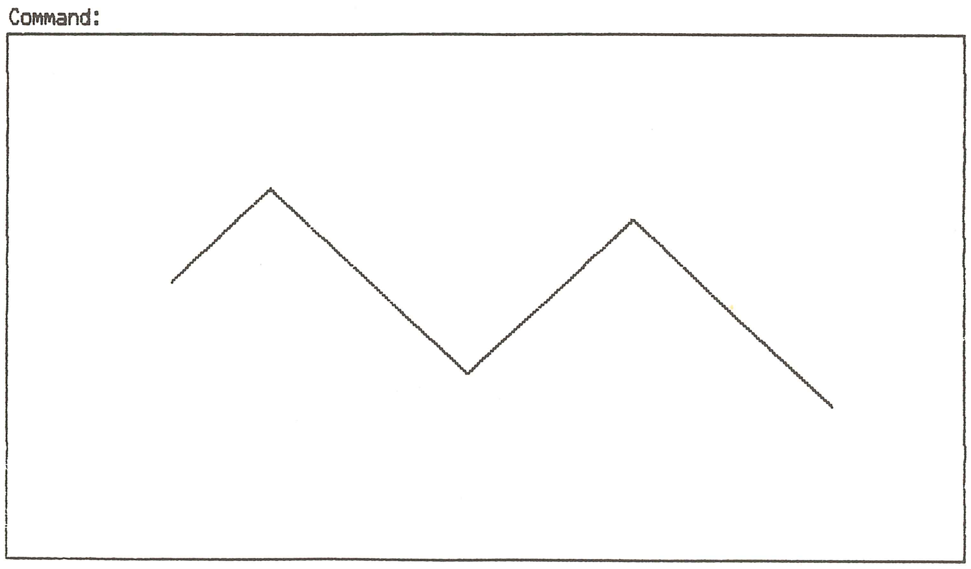
A point is a location on the screen. Each point corresponds to the intersection of a pixel on the horizontal line and a pixel on the vertical line. As GIGI can reference 768 pixels on the horizontal lines and 240 pixels on the vertical lines, a total of 184,320 points can be referenced in all by GIGI.

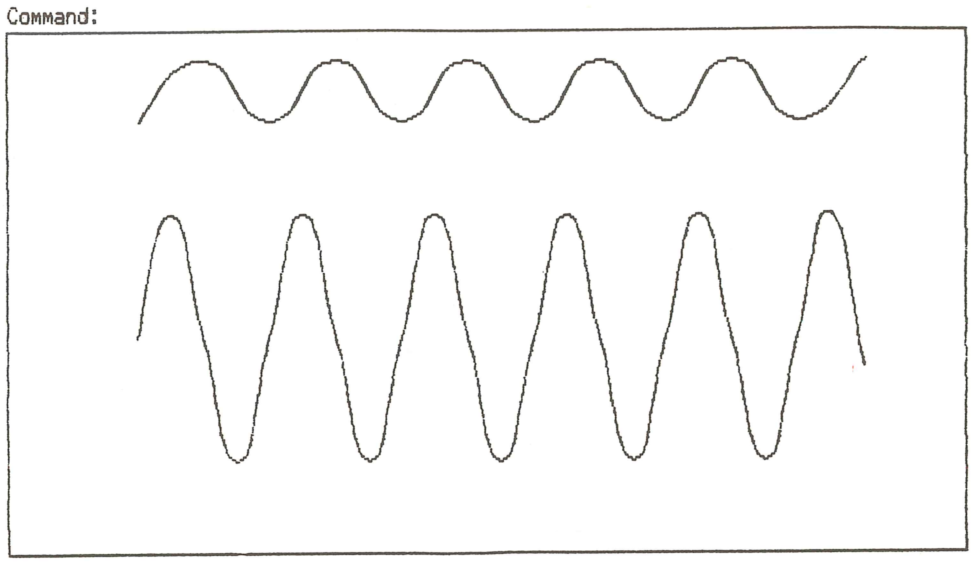
Points are displayed when you use either writing or screen functions. Points can be displayed in colors, which you can select with the Graphics Editor attributes. The Graphics Editor uses default colors of white for writing and black for background, if you do not specify other colors.

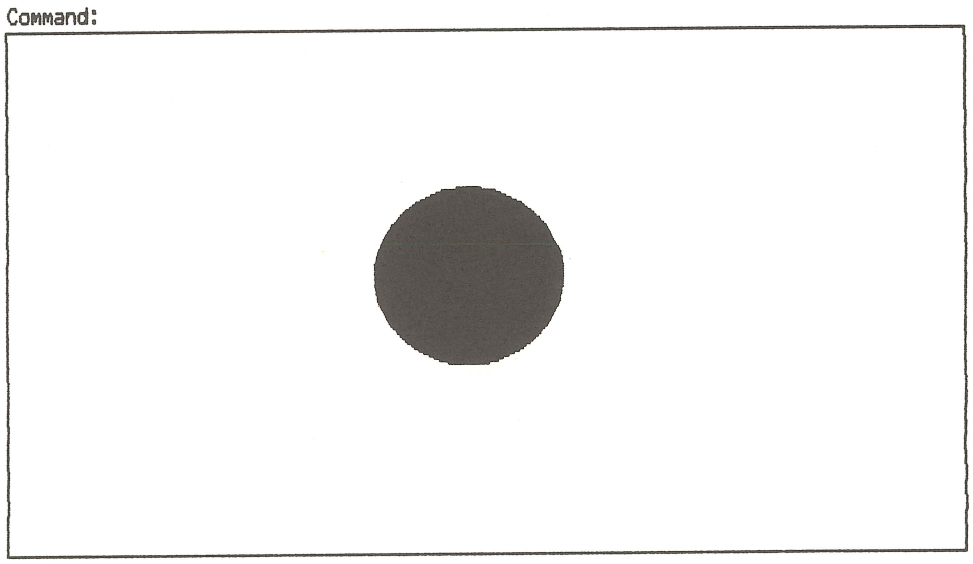
Objects

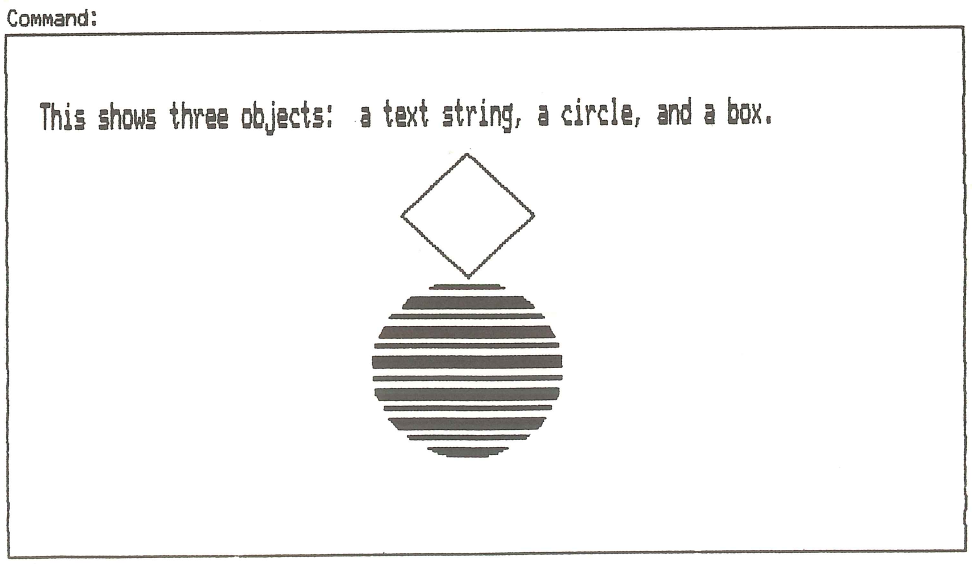
The Graphics Editor actually creates pictures by displaying objects, rather than individual pixels. An object is the smallest element that you can manipulate with the Graphics Editor. An object is a distinguishable part of a picture, and consists of one or more ReGIS statements.

The objects you can manipulate with the Graphics Editor are straight lines (referred to as lines), curved lines (referred to as curves), boxes, circles, text strings, and erases (also referred to as pages). Each individual entity is considered to be an object. Examples of screen displays with single and multiple objects are shown below.


```
Command:
```







This shows three objects: a text string, a circle, and a box.

FUNCTIONS

The Graphics Editor provides functions that allow you to easily display objects interactively without using ReGIS statements.

Four of the Graphics Editor functions draw basic graphic elements; these are the LINE, CURVE, CIRCLE, and BOX functions. As their names indicate, these functions draw lines, curves, circles, and boxes respectively.

Another Graphics Editor function, the TEXT function, permits the incorporation of text characters in pictures. The TEXT function provides access to the standard ASCII character set and to user-created character sets, such as APL or Greek characters or math symbols. User-created character sets are created with the Character Set Editor or with ReGIS and a standard text editor.

The remaining Graphics Editor function, ERASE, creates a blank page. When the ERASE function is used, it clears the screen of the existing display, while still retaining the previous picture within GIGI itself.

ATTRIBUTES

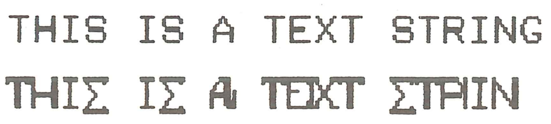
The Graphics Editor provides a wide variety of attributes you can use to enhance your pictures. Attributes such as Slope, Italic, and Height let you change the appearance of text characters. The Font attribute provides a means of using alternate character sets. You can select the color in which objects are written, as well as the background color of the screen. You can use shading in various patterns (including solid, dotted, dot-dashed, and dashed) to highlight objects. Blinking can be turned on or off for objects.

CHARACTER SETS

Character sets are sets of characters used with the TEXT function. GIGI can use up to four character sets, one of which is the ASCII character set and is always loaded, and three other user-defined character sets. Characters sets are the descriptions of characters. Alternate character sets are the part of GIGI memory into which you can load the character sets.

Using the Character Set Editor or ReGIS, you can create a wide variety of characters. The characters can be letters or any kind of symbol. When these characters are created, they are associated with the keys on the main keypad. To display the user-defined character sets, you use the corresponding keys on the keypad, after loading the character sets and selecting the appropriate alternate character set. By using the text function and entering characters, you enter whatever character you defined with the Character Set Editor.

The result of using a character set is that the Graphics Editor will display the character that is represented by the ReGIS definition for that character. The example below shows two lines of text. The first line is ASCII text. The second line shows a character set that contains descriptions of Greek characters.



DRAWING PICTURES

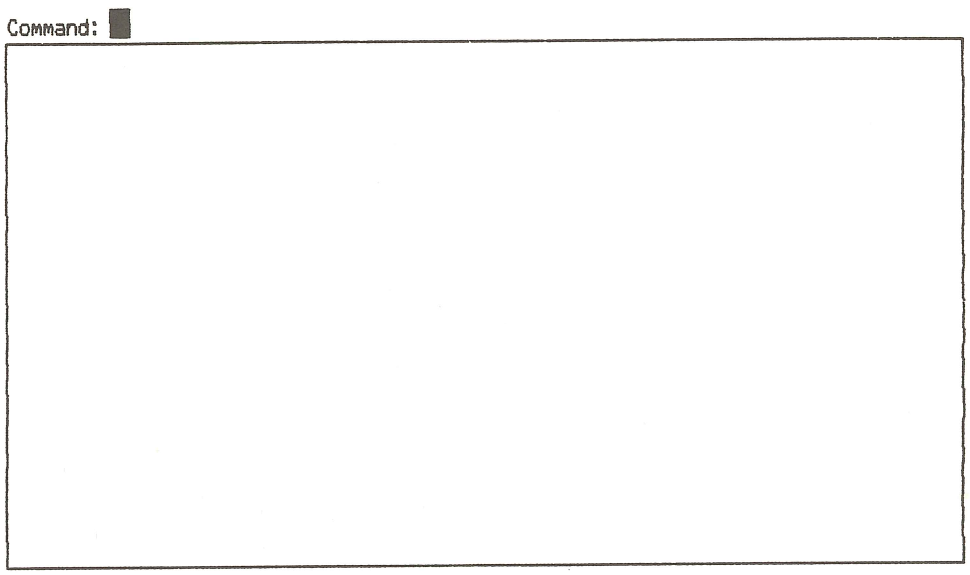
The Graphics Editor provides easy access to the drawing capabilities of GIGI and ReGIS. To begin with the Graphics Editor, it is suggested that you start by drawing simple pictures and using a minimum of attributes. After trying some simple pictures, try using some of the other features of the Graphics Editor that give you access to the more complex features of GIGI.

It is suggested that you read Chapter 3 (Graphics Editor Keyboard), Chapter 4 (Files and Saving Pictures), and Chapter 5 (Drawing Pictures) first, as these chapters explain the use of the Graphics Editor.

USING THE GRAPHICS EDITOR

The Graphics Editor is interactive. To draw pictures you use keypad commands and typed commands. Keypad commands are commands that are executed by merely pressing a single key. Typed commands are executed by typing the name of the command. Each command can be abbreviated to the first letter of the command.

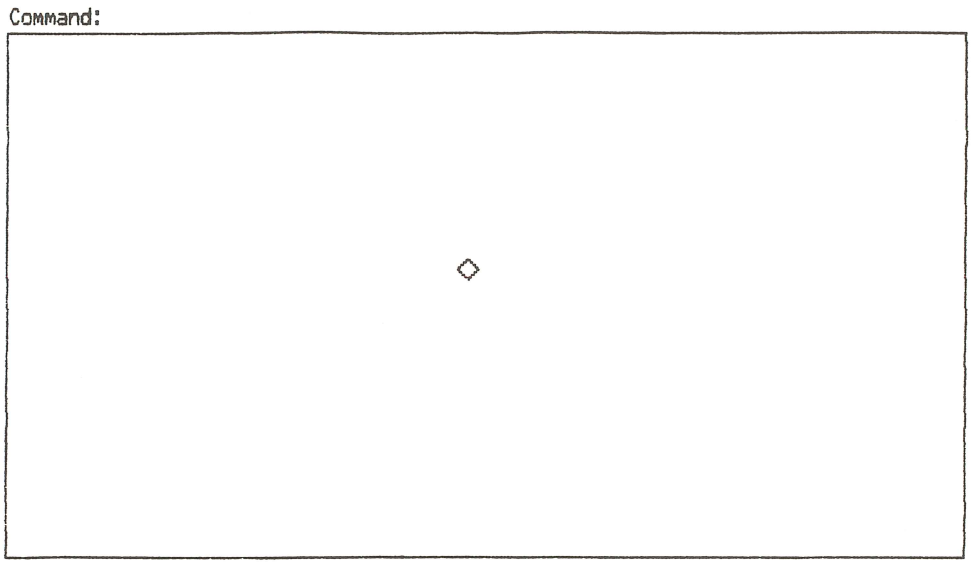
When you start the Graphics Editor, the screen appears as shown below.



```
Command: █
```

The rectangular cursor, which is referred to as the cursor, will be blinking beside the word Command. When the cursor is blinking at the top of the screen, the Graphics Editor is at command level. At command level, you can use any of the typed commands and any of the keys that perform functions.

After using certain keys, another cursor is displayed within the picture frame the Graphics Editor uses. This cursor, which is shaped like a diamond, is the graphics cursor. When the graphics cursor is displayed, the Graphics Editor is at function level. The example below shows the screen display when the Graphics Editor is at function level.


```
Command:
```

The command and function levels are described in more detail in Chapter 3.

HELP

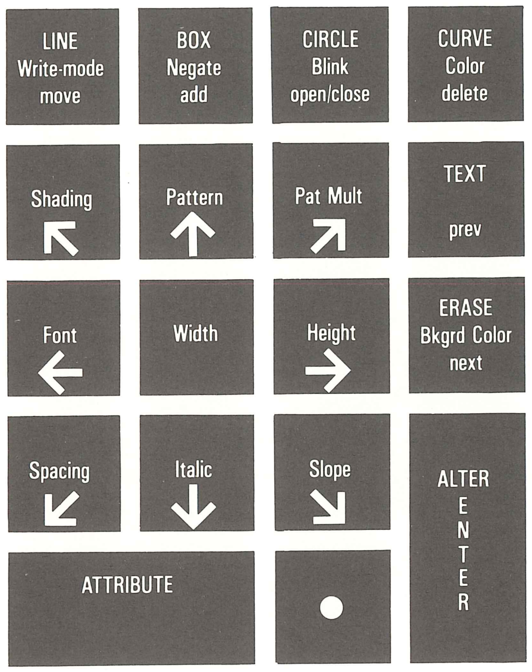
The Graphics Editor has a help key. For a display of the typed commands you can use, press the Width key. The Graphics Editor displays a list of the commands and the syntax for each command.

ERRORS

The Graphics Editor provides a large number of functions that are available with just a few keys. As the next chapter describes, the performance of the key depends on what key was pressed previously. If you press the wrong key, nothing happens in most cases. However, if the display is different from what you think it should be, you can easily delete the object by pressing the RETURN key on the main keypad. The Graphics Editor will return to command level, and the object you were drawing will be deleted. You can then try again.

3 GRAPHICS EDITOR KEYPAD

The Graphics Editor uses the keys on the auxiliary keypad to draw and modify pictures. These keys perform multiple functions. The keypad overlay used with the Graphics Editor fits over the auxiliary keypad and shows the names of the keys. The following diagram shows the keypad overlay.



These keys provide four levels of operation: function, attribute, interaction, and control. Function, attribute, and interaction keys are grouped together on the same physical keys, while control keys are grouped only with other control keys.

Function keys draw objects: lines, boxes, circles, curves, text, and screen erases.

Attribute keys change settings of attributes: writing mode, negative writing, blinking, object color, pattern, pattern multiplier, shading, font, width, height, screen color, spacing, italics, and slope.

Interaction keys allow interaction with individual objects. Interaction keys move the graphics cursor from object to object and from point to point on single objects, and manipulate marks and individual text characters.

Control keys provide the means for changing levels and for performing special functions.

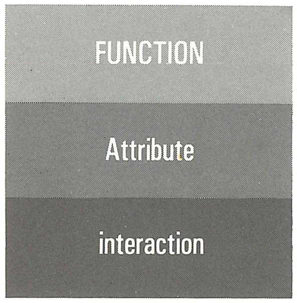
To distinguish the types of keys, this manual uses the following conventions:

Level	Convention	Example
Function	UPPERCASE	BOX
Attribute	Initial Capital	Shading
Interaction	lowercase	delete
Control	UPPERCASE	ENTER

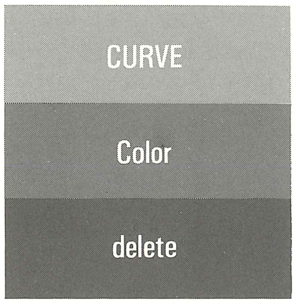
The different uses of each key are determined by the sequence in which you use them. To draw and modify pictures, keys must be used in the correct sequence.

Using keys in the wrong sequence does not destroy your picture. In most cases, the Graphics Editor either does nothing or it has the means to allow you to easily correct the problem. In cases of more drastic operations, such as deleting one or more objects, the Graphics Editor prompts you to confirm your choice before it deletes the object or objects.

These levels of operation are grouped in the same arrangement on each key. The example below shows the grouping of the function, attribute, and interaction levels on keys:



The following example shows the key on the top right of the auxiliary keypad. The CURVE key is a function key, the Color key is an attribute key, and the delete key is an interaction key.



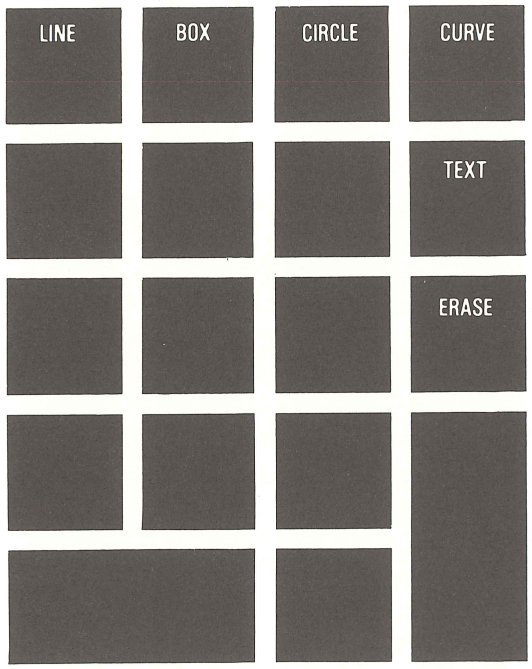
Each name shown on the keypad is the name of a key. For example, this physical key is called the CURVE key, the Color key, and the delete key.

Since the sequence of use determines the level of the key, you will sometimes use the same physical key consecutively. The level at which the physical key performs will be determined by what you did previously.

FUNCTION KEYS

The top level is the function level. Using this key specifies a function.The Graphics Editor must be at command level for the key to be interpreted as a function key. The Graphics Editor is at command level when the cursor is blinking beside the command prompt at the top of the screen. An example of the cursor at command level is shown in Chapter 2.

Functions indicate what type of object or image will be drawn on the screen. The functions are lines, boxes, circles, curves, text, and screen erase. The diagram below shows the function keys.



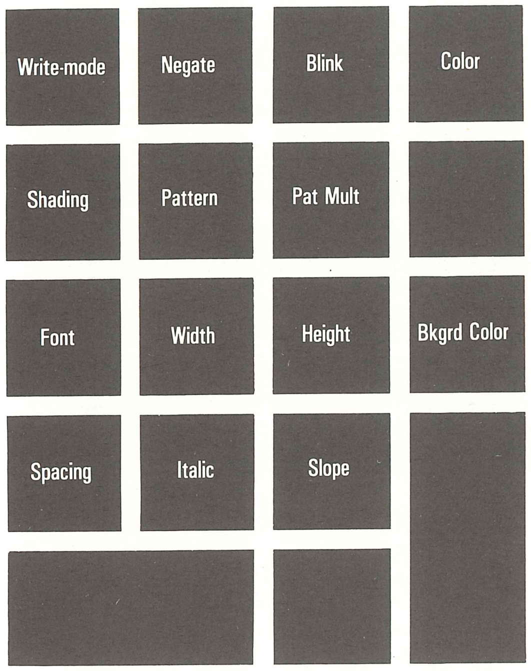
ATTRIBUTE KEYS

The second level specifies an attribute. You can use the attribute keys at either of two levels: command level or function level. You are at command level when the cursor is blinking beside the command prompt. You are at function level after having pressed a function key. At function level the graphics cursor is blinking on the screen.

To choose an attribute at either level, you first press the ATTRIBUTE key (in the lower left corner of the auxiliary keypad). You then select an attribute by pressing the appropriate attribute key, such as Shading. Continue pressing the attribute key until the desired setting is displayed, and then press the ENTER key. Pressing the ENTER key changes the setting to the setting displayed on the screen.

To return to either command or function level without changing the setting, press the RETURN key on the main keypad.

The attribute keys are shown in the following diagram.



INTERACTIVE KEYS

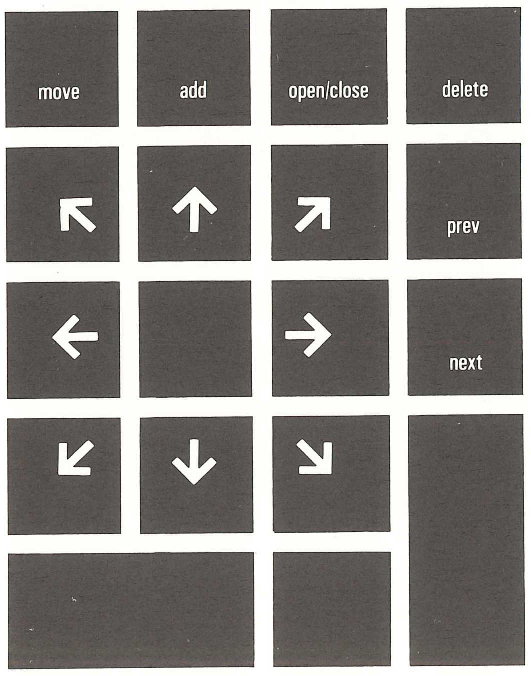
The third level is the interactive level. To be at interactive level, you must have already pressed a function key. Once you have selected a function key and are at function level, the interactive keys are active. At function level, the graphics cursor is blinking on the screen.

At function level, you can use the interactive keys in any sequence with each other and with the attribute keys.

The interactive keys on the top row have special functions when used with circles and boxes. These are described in Chapter 5, Drawing Pictures, and in Chapter 7, Auxiliary and Main Keypad—Reference.

To leave interactive level and return to command level, press either of the following keys: the ENTER key on the auxiliary keypad or the RETURN key on the main keypad. The ENTER key returns you to command level and saves the object. The RETURN key returns you to command level, but does not save the object.

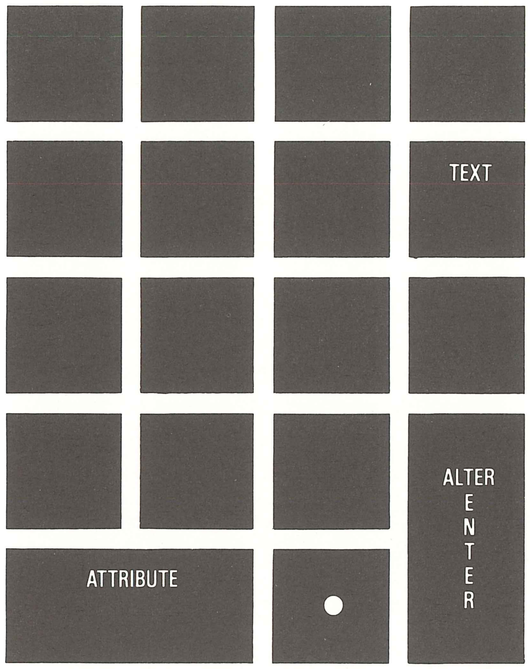
The interactive keys are located on both the auxiliary keypad and the main keypad. The keys on the main keypad are the arrow keys. These keys perform the same functions as the directional keys on the auxiliary keypad. However, the arrow keys make changes of two pixels, and the directional keys make changes of five pixels.



CONTROL KEYS

The control keys provide control functions for the Graphics Editor. Control keys are grouped together on the bottom keys of the auxiliary keypad. Control keys provide ways of changing levels while using the Graphics Editor. The control keys are active at both the command level and the function level.

The diagram below shows the control keys.



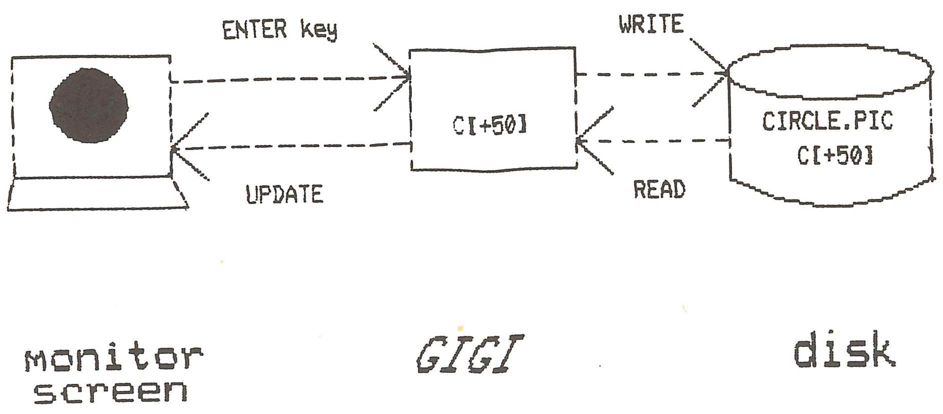
4 FILES AND SAVING PICTURES

The Graphics Editor represents pictures in three ways: as an image on the monitor screen, as a sequence of ReGIS commands in the internal file copy used by the Graphics Editor, and as a sequence of ReGIS commands in a file on disk.

This chapter describes the use of files with the Graphics Editor: creating, writing, reading, and editing files; accessing objects in files; and using picture file formats.

USING FILES WITH THE GRAPHICS EDITOR

The use of any function key (and any associated attribute keys) displays an object on the monitor screen. The ENTER key is used to save the object: it writes the ReGIS commands describing the object (and its attributes) to the internal file copy used by GIGI. The WRITE command writes the internal file copy to disk in the file with the specified file name. After the file is written to disk, the READ command is used to read the file into GIGI; GIGI then has an internal file copy of the file. The UPDATE command then graphically displays the contents of the internal file copy. The example below shows this process.



In this diagram, the CIRCLE key was used to display a circle on the monitor screen. Nothing is written to the internal file copy or to the file on disk.

The use of the ENTER key writes the ReGIS command for a circle to the internal file copy. This does not write the ReGIS command on disk.

The subsequent use of the WRITE command and a specified filename, CIRCLE.PIC, then writes the ReGIS command in ASCII format in the file CIRCLE.PIC on disk.

After a file has been written to disk, it can be read back into GIGI with the READ command. In this example, the statement READ CIRCLE.PIC reads the file back into GIGI, making the file available for displaying and editing. Use the UPDATE command at this point to display the contents of the file.

CREATING FILES

To create a file with the Graphics Editor, use the WRITE command and specify a filename. If the file does not yet exist on disk, the WRITE command creates a file with the specified name and writes the current internal file copy to disk. If a file of the specified name already exists, the WRITE command writes the internal copy of the file to the specified file on disk. The new copy of the file overwrites the old copy of the file.

WRITING FILES

To write the internal file copy to disk, use the WRITE command and specify the file name.

Before an image displayed on the screen can be written to disk, the image must be saved in the internal file copy. To save the image in the internal file copy, use the ENTER key. Displaying the object does not write the ReGIS commands for the object to the GIGI file copy. If you return to command level without using the ENTER key to save the object in the internal file copy, the object is lost and must be redrawn with the function keys if it is to be displayed again.

Using the ENTER key to save an attribute setting does not write the display to the GIGI file copy.

After a file has been written to disk, you can access the disk file with the Graphics Editor or with any of the standard systems methods, such as the TYPE command or a standard text editor. To access the disk file with the Graphics Editor, use the READ and UPDATE commands after starting the Graphics Editor. To access the disk file with a TYPE command, specify the TYPE command and the file name. The TYPE command types the disk file, which appears as a string of ReGIS commands. Using a standard text editor allows you to edit the ReGIS commands in the disk file in the same way that a text file would be edited.

READING FILES

The READ command reads copies of files from disk to GIGI. A copy of the file remains on disk, and another copy of the file is read into GIGI. After the file copy is read into the buffer, the file can be displayed with the UPDATE command. Using the UPDATE command displays the objects in the same order the objects were originally drawn.

EDITING FILES

The Graphics Editor can edit existing picture files that reside on disk. To do this, use the READ command to read a copy of the disk file into GIGI. The Graphics Editor can then be used to edit the picture display of the internal file copy. After the file has been edited, the file can be written to disk with the WRITE command.

To combine multiple existing files into one file, use the READ command to read each of the desired files into the Graphics Editor.After all the files have been read in, use the WRITE command to write the new file. This file then contains all the files that were read into the Graphics Editor. The pictures in the new file will be in the same order as they were read into the Graphics Editor.

PICTURE FILE FORMATS

Each object saved in the internal file copy is assigned a relative number based on the order in which it was read into the file. The first object drawn is the first object in the file; the second object drawn is the second object in the file; and so on, to the end of the file.

For example, drawing a circle, a box, and a curve with the function keys and then pressing the ENTER key would write three objects to the internal file copy. If the objects are drawn in that order, they have the following sequential arrangement:

- circle
- box
- curve

In this example the circle is object 1, the box is object 2, and the curve is object 3.

Accessing Objects in Files

Each object in the GIGI file copy can be accessed directly in one of three ways:

- using the cursor to specify the current position
- using a label to indicate a range
- specifying a range of one or more objects

Using the current position

Using the current position specifies the object where the graphics cursor is currently positioned. A • (a period on the main keypad) is used with commands to indicate the current position. You can move the graphics cursor to any object in the picture and thus use any object for the current position. To move the graphics cursor, first press the ALTER key. When the prompt appears, move the graphics cursor with the next or prev keys. The object on which the graphics cursor is blinking is the current position. When you return to command level (by pressing the • key on the alternate keypad) that object is still recognized by the Graphics Editor to be the current position for the graphics cursor.

Once you have selected the current position, you can use this as an argument for commands, such as the TILT and SCALE commands. Accessing an object by using the current position is useful for specifying one object to be affected by the command you are using.

Using a label

Using a label to indicate a range allows you to assign names to objects within the GIGI file copy. Three types of labels can be used: GROUP labels, LABEL labels, and NAME labels. A label can include one to ten alphanumeric characters, of which the first character must be alphabetic.

The GROUP label identifies a range of objects; the LABEL and NAME labels flag one object. GROUP labels apply to all objects within the designated range.

GROUP and LABEL labels are written to the file on disk with the WRITE command. NAME labels are not written to the file on disk.

Using ranges

Using sequential numbers or symbols allows you to specify a range of one or more consecutive objects in the GIGI file copy. You specify both the beginning and the ending objects of a range, separating them with a comma. The beginning and ending objects can be specified with object numbers or symbols. Both the beginning and the ending objects of the specified range are included in the range.

The Graphics Editor recognizes certain symbols:

•	current object
n	the relative number of an object
$	the last object in a picture
@	last address range used
*	all objects in a file (1,$)
labels	GROUP, LABEL, and NAME labels

The commands that use ranges and symbols are:

ALTER	READ
COPY	SCALE
DELETE	TILT
GROUP	UPDATE
MOVE	

With each of these commands, you can specify an address. Examples of using ranges with commands are shown below.

ALTER 1,10
This specifies the range of objects 1 through 10 inclusive to be used with the ALTER command. Until returning to command level, only objects 1 through 10 can be accessed.

MOVE @
This sets the range used in the last command as the range to be moved.

DELETE •
This deletes the object at the current position (where the graphics cursor is positioned).

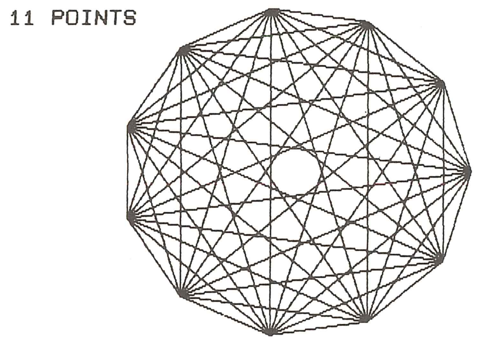
GROUP example 2,4
This assigns the GROUP label *example* to objects 2, 3, and 4.

SCALE example
This sets the range established for the label *example* to be scaled.

COPY $
This copies the last object in the file.

UPDATE *
This updates or displays every object in the file.

Picture files that have been written to disk contain sequences of ReGIS commands. For a detailed explanation of ReGIS and its use, refer to the GIGI/ReGIS Handbook. The example below shows a picture created with the Graphics Editor. The ReGIS statements for this file are in the file, PICTURE.PIC. Below the picture of PICTUR.PIC is a listing created with the Graphics Editor of the file PICTUR.PIC. This listing was generated with the TYPE command at system command level.



```
w(r,n1,p11111111(m1),s1[,260],m1,a0,i2)
t(a0,d0,s1,h2,i0)
w(v,n0,p1(m2),s0,i7)s(i0)
p[1,-1]c[0,0]
p[0,0]s(i0,e)
w(r,p11111111(m1),i4)
p[380,260]c[600,260]
w(s1[,260],i2)
p[380,260]c[580,260]
w(n1)
p[260,140]c(b)
  [500,380]
  [500,140]
  [260,380](e)
```

5 DRAWING PICTURES

The Graphics Editor provides functions for you to draw pictures using the alternate keypad, arrow keys on the main keypad, and typed commands.

You draw the pictures interactively using these keys and commands. The entire process of drawing and saving a picture requires the following steps:

1. Use of a function key to display the object.
2. Use of the attribute and interaction keys to modify the object.
3. Use of the ENTER key to save the object in GIGI.
4. Use of the WRITE command to write the file to disk.

As you are drawing and modifying pictures, you can use the UPDATE command to refresh the screen if it is necessary. Using the UPDATE command redraws the screen to show the picture as represented by the internal copy of the file.

This chapter describes the process of drawing pictures using the function, attribute, and interaction keys. Refer to Chapter 4 for information about saving pictures and using files.

GRAPHICS EDITOR FUNCTIONS

The first step in drawing pictures is to use the function keys. These keys draw objects, which are the basic elements of a picture. The objects you can draw with the function keys are:

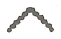
- lines
- curves
- circles
- boxes
- text
- erase (new pages)

The following sections describe each of these functions.

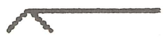
Lines

The Graphics Editor draws lines with the LINE key. A line is a straight geometric element generated by moving the graphics cursor and setting marks. A mark is a position on the screen and is represented with a carat, ^. Marks are placed with the add key.

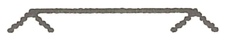
You begin by pressing the LINE key. After the prompt is displayed, you position the graphics cursor with the arrow or directional keys, and press the add key to place the mark. It looks like this:



After placing the first mark, use the arrow or directional keys to move the graphics cursor to the next position. As the graphics cursor moves along the screen, the Graphics Editor displays the line from the last mark to the cursor position. The screen display looks like this:



After moving the graphics cursor to the next position, use the add key to place the mark. Continue moving the graphics cursor and setting marks with the add key until the line is completed. The line and the marks now look like this:



To save the line, press the ENTER key. The example below shows the finished line. Using the ENTER key deletes the display of the marks.

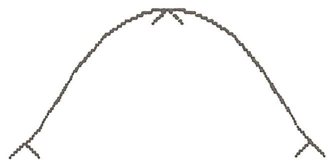
To draw two or more unconnected lines, repeat this process starting with the LINE key You must press the ENTER key between unconnected lines. After using the ENTER key, you then press the LINE key again and position the graphics cursor to begin the new line. Each line must be entered separately or else the Graphics Editor will draw one continuous line to connect all the marks.

Curves

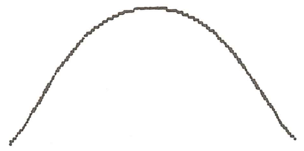
The Graphics Editor draws curves with the CURVE key. A curve is a curved geometric element generated by moving the graphics cursor and setting marks. GIGI interpolates curves; that is, given a sequence of locations, GIGI computes intermediate points and connects them with small vectors which fit an appropriate curve. The shape of the curve depends on the location of the marks. To draw a curve, you must set at least three marks, as the Graphics Editor draws a straight line for two marks.

To draw a curve, follow the same procedure as drawing a line, but start by pressing the CURVE key. When the Graphics Editor displays the prompt, position the graphics cursor at the beginning of the curve, and set the first mark with the add key. After setting the first mark, move the graphics cursor to the appropriate position, set a second mark, move the graphics cursor to the next position, and set the third mark.

As the graphics cursor moves along the screen, the Graphics Editor displays the line from the last mark to the cursor position. The Graphics Editor displays a straight line between the first two marks, and displays the curve after the third mark is placed. The screen display looks like this when three marks have been set:



Continue moving the graphics cursor and setting marks until the curve is complete. To save the curve, press the ENTER key. The example below shows the curve after pressing the ENTER key. The marks are no longer displayed.



To draw two or more unconnected curves, repeat this process. The ENTER key must be pressed after drawing each separate curve or else the Graphics Editor will connect each mark and draw one continuous curve.

Circles

The Graphics Editor draws circles and arcs with the CIRCLE key. You can modify the location and the size of the circle, as well as any of the circle's attributes. You can also modify the size, number of degrees, and location of arcs with the CIRCLE key.

To draw circles and arcs, first press the CIRCLE key. After the CIRCLE key is pressed, the keys on the top row of the auxiliary keypad assume special functions. These functions are:

move key	decreases radius by five pixels
add key	increases radius by five pixels

These keys perform the same functions no matter where the graphics cursor is positioned. Other keys used to modify circles and arcs perform different functions, depending on the position of the graphics cursor.

The graphics cursor can be placed in one of three positions: in the center of the circle (figure 1), on the circumference without a radius displayed (figure 2), and on the circumference with the radius displayed (figure 3). To move the graphics cursor to these various positions, use the next or prev keys. The example below shows the graphics cursor in the three possible positions.

Figure 1

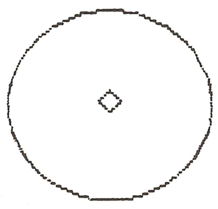


Figure 2

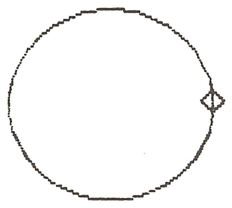
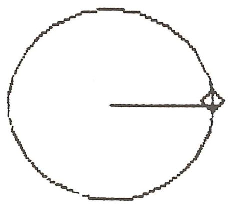


Figure 3



Modifying Circles

Circles can be made larger and smaller and can be moved to different positions on the screen.

Two methods can be used to change circle size. The simplest method to change the size of circles is to use the move key and the add key. The move key decreases the radius of the circle by five pixels; the add key increases the radius of the circle by five pixels. Using the add and the move keys changes only the size of the circle, the center of the circle remains in the same place.

The second method of changing circle size is to use the directional and arrow keys. This method also changes the position of the graphics cursor on the circumference.

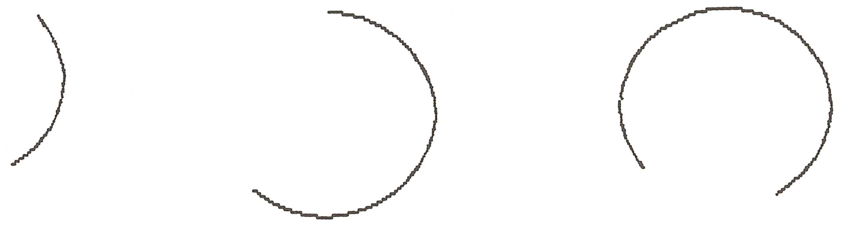
The graphics cursor must first be positioned (using the next or prev key) on the circumference without the radius displayed (figure 2). Pressing the arrow or directional keys at this point increases or decreases the size of the circle. The action of the arrow or directional keys depends on the location of the graphics cursor on the circumference and on the last key used. In some situations (based on the graphics cursor position and the previous sequence of keys that was used), no change occurs when the arrow or directional keys are pressed. If this happens, press another arrow or directional key. The directional keys cause changes of five pixels, and the arrow keys cause changes of two pixels.

This second method also moves the graphics cursor around the circumference of the circle. This is useful when creating arcs tilted to different angles.

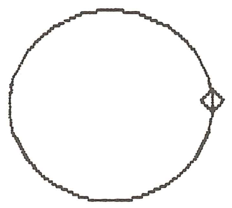
To change the circle position on the screen, first position the graphics cursor at the center of the circle (figure 1) by using the next or prev key. Then use the directional keys or the arrow keys to move the circle in the direction shown on the keycap or overlay. This changes only the location of the circle, not the size. The directional keys move the graphics cursor five pixels and the arrow keys move it two pixels.

Modifying Arcs

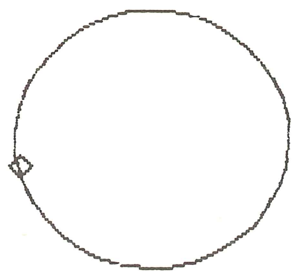
Arcs can also be drawn with the CIRCLE key. An arc is a section of the circumference of a circle. Examples of arcs of different degrees are shown below.



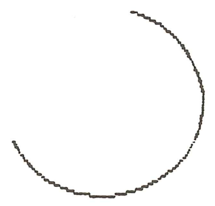
To create an arc, first use the CIRCLE key to draw a circle. Then, using the next or prev key, position the graphics cursor on the circumference of the circle without the radius displayed:



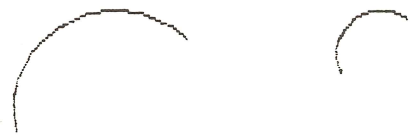
The next step is to define the arc itself. To position the graphics cursor where the arc will begin, use the directional or the arrow keys to move the graphics cursor around the circumference of the circle. The circle size changes as you do this. It is important to position the graphics cursor even if the circle size is not the size you want. You can change the size after you create the arc. The example below shows the circle with the graphics cursor placed in a different position.



When the graphics cursor is positioned where the arc is to begin, press the open/close key to decrease the arc. Use the delete key to increase the arc. The open/close and delete keys modify the arc by five pixels. The example below shows the same circle after the open/close key was used.



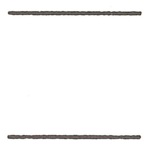
When the arc is the right number of degrees, use the arrow or directional keys to modify the size of the radius of the arc. If you want to change the size of the arc at this point, use the move key to decrease the arc radius by five pixels, or the add key to increase the arc radius by five pixels. The example below shows two arcs. The arc on the right is a copy of the arc on the left. The radius of the arc on the right was decreased with the move key.



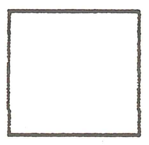
Boxes

The Graphics Editor draws boxes with the BOX key. Using the BOX key displays a box on the screen. After it is displayed, you can modify the size and location with the interaction keys. The Graphics Editor displays the top and bottom lines while drawing the box, and displays the complete box when the box is saved with the ENTER key.

To draw a box, first press the BOX key. When the box is first displayed, the graphics cursor is positioned on the upper left corner of the box. Only the top and bottom lines of the box are displayed while the box is being drawn or modified. At this point the box looks like the example shown below.



When the box is saved with the ENTER key, all four sides of the box are displayed. The box then looks like the following example.



Two methods of using the interaction keys can be used to change the size and the position of a box. One method involves using the top row of keys; the other method involves using the next and prev keys to position the graphics cursor, and the arrow and directional keys to modify the size and location of the box.

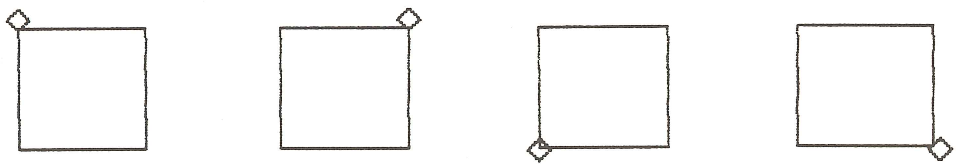
The first method of modifying a box is to use the top row of keys on the auxiliary keypad. The keys have the following effects:

move key	decreases width five pixels
add key	increases width five pixels
open/close key	decreases height five pixels
delete key	increases height five pixels

These keys can be used after pressing the BOX key. The example below shows a box as it originally appears on the screen, and then after the box has been modified with the add key.

The second method involves the use of the next or prev key, and the arrow and directional keys. The position of the graphics cursor on any corner determines the effect of the interaction keys on the box. When the graphics cursor is on the upper left or the lower right corner, you can change the box location. When the graphics cursor is on the upper right or the lower left corner, you can change the size of the box.

You can move the graphics cursor to any of the four corners with the next and prev keys. The next key moves the graphics cursor in a clockwise direction and the prev key moves the graphics cursor in a counterclockwise direction. The example below shows boxes with the cursor in each corner position.

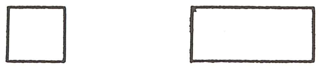


To move the box, first position the graphics cursor on either the upper left or the lower right corner. Then use the directional or arrow keys. These keys move the graphics cursor and the box in the direction indicated on the keycap. The box does not change size.

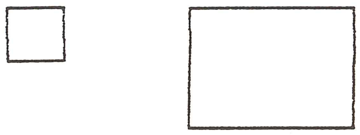
To change the size of the box, position the graphics cursor on either the upper right or the lower left corner with the next or the prev key.

Positioning the graphics cursor on the upper right corner allows you to modify the top line and the right side of the box. When positioned there, the directional and arrow keys move the graphics cursor in the direction indicated on the key. Moving the graphics cursor in a given direction moves the corner of the box in the same direction. For example, moving the graphics cursor to the top of the screen causes the top of the box to move toward the top of the screen. The lower left corner remains fixed.

The example below shows two boxes. The first box shows the box as it appears on the screen after pressing the BOX key. The second box is created by positioning the cursor on the upper right corner with the next key, and using the right arrow key. This increases the width of the box.



Positioning the graphics cursor on the lower left corner modifies the bottom and left lines of the box. Using the arrow and directional keys to move the graphics cursor also moves the lower left corner of the box in the same direction. As the graphics cursor moves, the lower left corner of the box moves. As the lower left corner moves, the bottom and left side of the box change appropriately. The upper right corner remains fixed. The example below shows two boxes. The first box shows how the box first appears on the screen. The second box is created by positioning the cursor on the lower left corner with the prev key, and using the ↙ key.



Text

The Graphics Editor can include text in pictures. Text can consist of any of the 95 printing characters on the main keypad, including letters, numbers, symbols, and spaces. The Graphics Editor also has the ability to use fonts created with the Character Set Editor or ReGIS.

Text Cells

Text characters actually consist of the displayed character and a text cell, in which the character is enclosed. The text cell surrounding the character is modified by some of the Graphics Editor attributes, and determines the spacing between characters. Each text cell is eight pixels by ten pixels when the default character size is used. The example below shows the character E in a text cell.

You can control the display of the text cell surrounding the character by choosing certain combinations of attributes and objects. Attributes that affect this aspect of the text display are Write-mode and Negate. This display is also affected by the attributes assigned to the underlying objects, such as background color, writing patterns, and shading.

Text Strings

Text can be arranged anywhere on the screen into a text string. A text string corresponds to an object; that is, all characters typed in between using the TEXT key and the ENTER key are considered to be a text string. Below is an example of a standard ASCII text string created with the GIGI text function.

```
****************************************************************
*                                                              *
*    ON THE SCREEN YOU WILL SEE                                *
*         A TEXT STRING IN ASCII                               *
*                                                              *
****************************************************************
```

Characters can be arranged to create text strings of varying appearances. The example below shows characters arranged around the screen, rather than in a horizontal line. This also is considered a text string, because all of the characters were entered between using the TEXT key and the ENTER key.

```
D
  I
    A
      G
        O
          N
            A
              L
```

Alternate Character Sets and Font Files

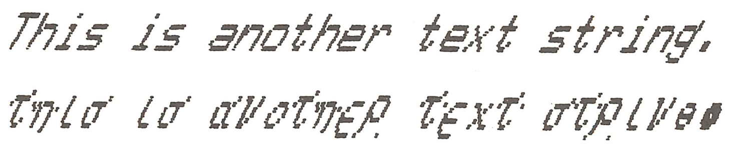
Text is displayed as standard ASCII characters, unless you are using GIGI's alternate character sets. The GIGI alternate character set allows you to use the main keyboard to display characters or symbols that you have designed with the Character Set Editor. The text string shown below uses APL characters that were created with the Character Set Editor.

```
⍺⊥∩⌊∊_∇∆⍳∘'⎕|⊤○*?⍴⌈∼↓∪⍵⊃↑⊂
```

To use alternate character sets, a font file must first be created with either the Character Set Editor or ReGIS. Font files are files containing the ReGIS statements to define user-created characters. For example, one font file might contain the ReGIS statements for math symbols, and another font file might contain the ReGIS statements for GREEK letters. For information about creating a font, refer to the Character Set Editor Manual; for information about using ReGIS, refer to the GIGI/ReGIS Handbook.

GIGI has one standard character set and three alternate character sets. The ASCII font is always loaded in character set 0. The three alternate character sets can simultaneously be loaded with one font in each alternate character set. The Graphics Editor FONT command loads the font file into the specified alternate character set. To enable the use of the appropriate alternate character set after the font has been loaded, use the Font key.

After the font has been loaded and the alternate character set has been selected, use the TEXT key to display the characters. The characters appear according to the definition in the alternate character set. The example below shows two text strings. The first string uses the default character set which always contains the ASCII character set. The second string uses a font file that was loaded into alternate character set 1. The same keys on the main keypad were used to create both text strings.



This series of steps is summarized as follows:

1. Use the Character Set Editor or ReGIS to create a font file.
2. Run the Graphics Editor and use the FONT command to load the font file into an alternate character set.
3. Use the Font key to select the alternate character set with the font file.
4. Use the TEXT key to begin entering text.

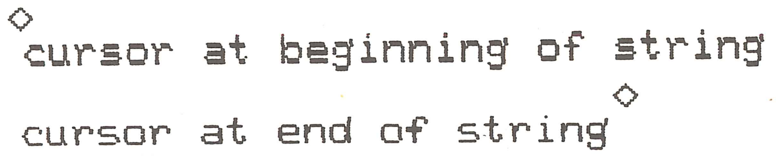
Entering and Moving Text

To enter text, first press the TEXT key. You can then use any of the 95 printing characters on the main keypad. To save text in the GIGI file copy, use the ENTER key. The Graphics Editor treats all text typed between pressing the TEXT key and the ENTER key as one object, even though it may consist of several characters.

While typing text, the interaction keys (the next key, the prev key, the arrow and the directional keys) can be used to position and modify the characters.After pressing the TEXT key, use the arrow or directional keys to position the graphics cursor where the text should begin. Using arrow or directional keys after text is typed moves the entire string of text typed since the TEXT key was last pressed, regardless of where the cursor is positioned on the text string.

To change the graphics cursor position after typing a string of text, use the next and prev keys. The next key moves the graphics cursor toward the end of the string. The end of the string is considered to be the right and the bottom of the screen. The prev key moves the graphics cursor toward the beginning of the string. The beginning of the string is always considered to be the left and the top of the screen. The graphics cursor moves one character at a time. The order in which the characters are typed does not affect the direction in which the next and prev keys move the graphics cursor.

The example below shows two strings of text. In the first string, the graphics cursor is positioned at the end of the string. In the second string, the graphics cursor is positioned at the beginning of the string, having been moved there with the prev key.

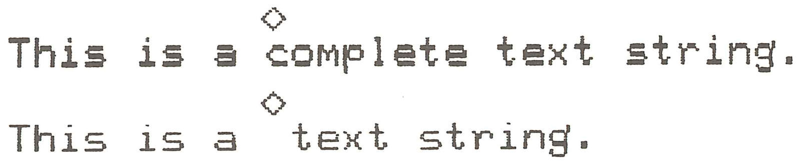


```
◇
cursor at beginning of string
                             ◇
cursor at end of string
```

Deleting Characters

To delete characters, use the delete key on the alternate keypad. The delete key deletes the character on which the graphics cursor is positioned. After a character is deleted, the remaining characters are shifted to fill the space occupied by the deleted character. Successive uses of the delete key remove characters to the right of the current position.

The example below shows two strings of text. In the first string, the cursor is positioned in the middle of the line. The second string shows the successive use of the delete key.



```
          ◇
This is a complete text string.
          ◇
This is a  text string.
```

Erases

The Graphics Editor creates new pages with the ERASE function. The screen appears to be erased as the new page is created. It is the same as putting one piece of paper on top of another. The first page remains the same; any drawings or text on the paper remain on the paper, although they are not visible. The top piece of paper or new page can then be used for writing and drawing. The ERASE function does the same thing. It creates a new page on which you can create pictures. The previous page or pages are no longer visible, although they still exist in the internal file copy.

ATTRIBUTES

To enhance the objects drawn with function keys, the Graphics Editor provides a variety of attributes for which you can select settings. Each attribute has a default setting used by the Graphics Editor unless you explicitly select another setting.

The table below shows the attributes that can be used with each function:

CIRCLES BOXES	LINES CURVES	TEXT	ERASE	Default
Bkgrd Color	Bkgrd Color	Bkgrd Color		black
Blink	Blink	Blink		off
Color	Color	Color		white
		Font		ASCII
		Height		1
		Italic		0
Negate	Negate	Negate		no
	Open/Close			open
Pattern	Pattern	Pattern		solid
Pat Mult	Pat Mult	Pat Mult		medium
Shading	Shading			off
		Slope		0
		Spacing		text
		Width		1
Write-mode	Write-mode	Write-mode		replace

Attribute Descriptions

The following sections describe the attributes and indicate which functions can be used with them. The only function not affected by attributes is the ERASE function.

Bkgrd Color

Bkgrd Color sets the background color; the background color is also referred to as the screen color. The background colors are black, blue, red, magenta, green, cyan, yellow, and white.The default background color is black.

Bkgrd Color can be used in conjunction with any of the function keys.

Blink

Blink causes objects to blink. Blink can be either off or on; the default is to be off. When blinking is on, the entire object blinks. If the object is shaded, the shaded area also blinks.

Blink can be used with the CIRCLE, BOX, LINE, CURVE, and TEXT functions.

Color

Color is the writing color used to draw the object. The writing coloris used to draw the outline of the object, as well as to do the shading if shading is turned on. You can select any one of the following colors: null, black, blue, red, magenta, green, cyan, yellow, or white. The default color is white.

Null means that the background color shows; no other writing color is used. If new pages have been entered with the ERASE function or if writing modes other than replace are used, the background color may be different from what it appears to be.

Color can be used with the CIRCLE, BOX, LINE, CURVE, and TEXT functions.

Font

Font specifies which of the character sets will be used by the Graphics Editor. GIGI has four character sets: one in which the standard ASCII character set is always loaded and three others in which you can load font files. The alternate character sets are empty unless a font file is read into them. To read a font file into the alternate character set, use the FONT command.

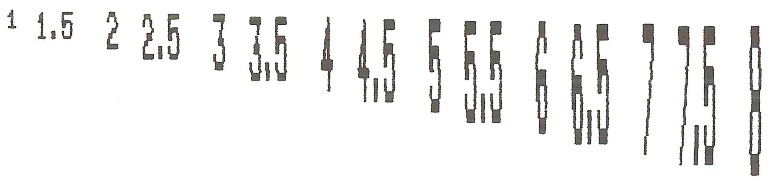
Font is used with the TEXT function.

Height

Height specifies the height of characters. The width and spacing is not affected. The character heights, which are incremented in steps of .5, range from 1 to 8.

Height is used with the TEXT function.

Below are examples of the height sizes.



Italics

Italics slants the characters. Characters can be slanted at the following degrees of tilt: −27, −45, +27, +45, and 0. Using Italics with Height can cause adjacent characters to overlap.

Italics is used with the TEXT function.

Below are examples of the Italics settings.

This shows 0 degrees slant.

This shows '27 degrees slant..

This shows '45 degrees slant..

This shows +45 degrees slant.

This shows -27 degrees slant.

Negate

Negate is negative writing. The background color is changed to the writing color and the writing color is changed to the background color.

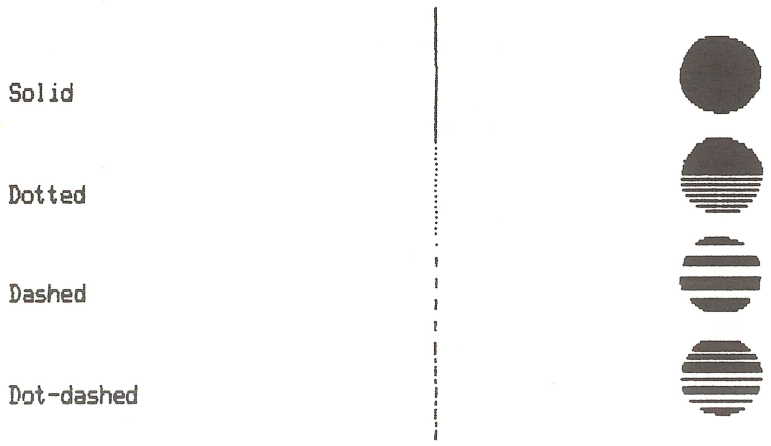
Negate can be used with the CIRCLE, BOX, LINE, CURVE, and TEXT functions.

Pattern

Pattern sets the pattern used for writing for both the outlines and shading of the objects. The patterns are solid, dotted, dashed, and dot-dashed. The default pattern is solid.

Pattern can be used with the LINE, CURVE, CIRCLE, and BOX functions.

Examples of each writing pattern are shown below.

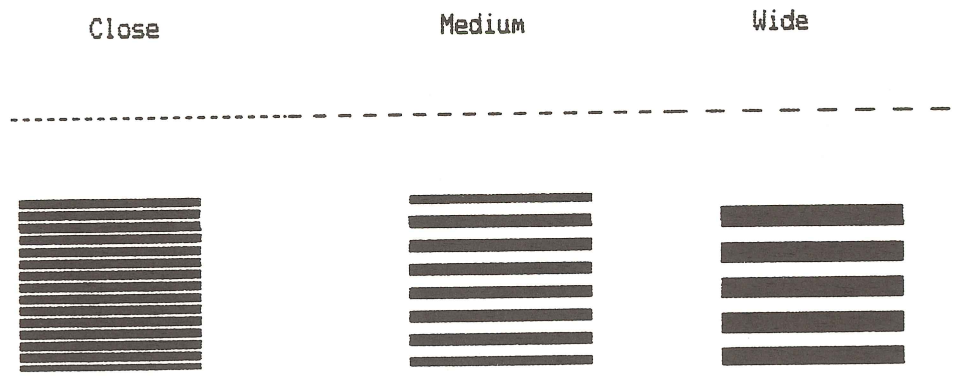


Pat Mult

Pat Mult (Pattern Multipler) multiplies or sets the size of the pattern. The pattern multiplier sizes are: close, medium, and wide. The patterns affected by the Pat Mult key are dotted, dashed, and dot-dashed. The Pat Mult modifies both outlines of objects and the shading of objects.

Pat Mult can be used with the LINE, CURVE, CIRCLE, and BOX functions.

The examples below show use of the Pat Mult keys.

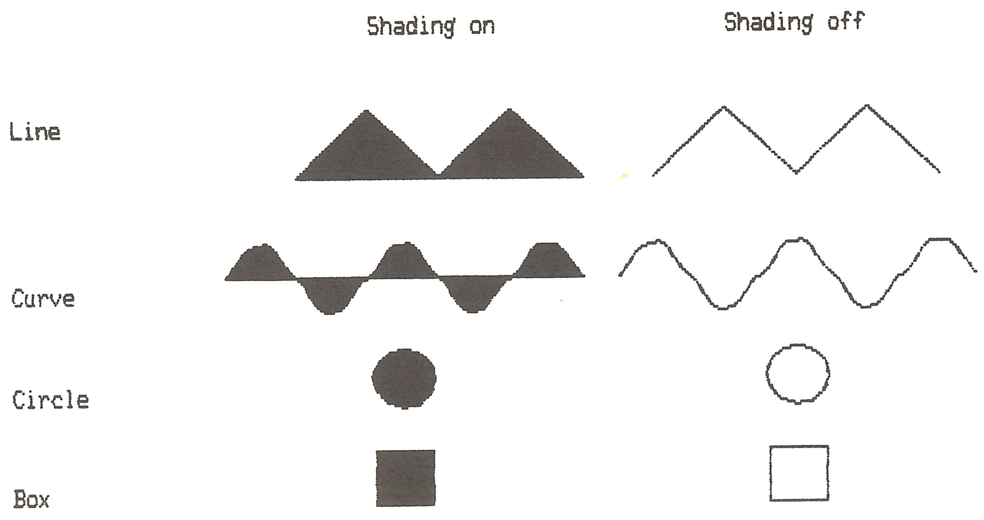


Shading

Shading fills in the specified object or objects with the writing color. Shading can be either on or off. The default is for shading to be off.

Shading can be used with the LINE, CURVE, CIRCLE, and BOX functions.

The examples below show use of the Shading key.

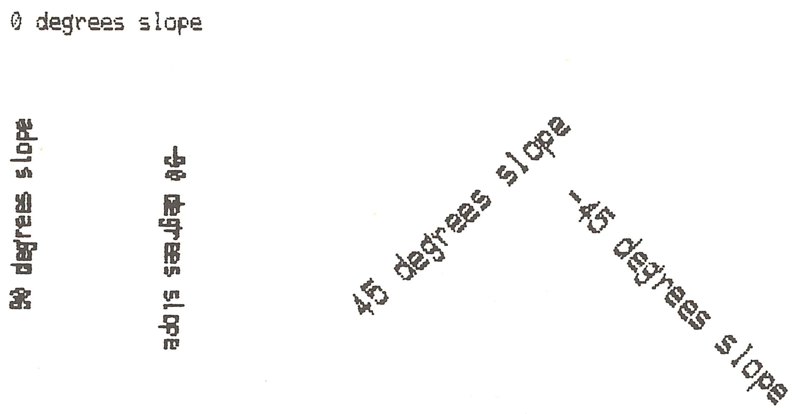


Slope

Slope controls the angle at which text is written. The angles at which text can be written are: 0, 45, 90, −45, and −90 degrees. The default is 0 degrees.

Slope is used with the TEXT function.

The examples below show text written with the different slope settings.

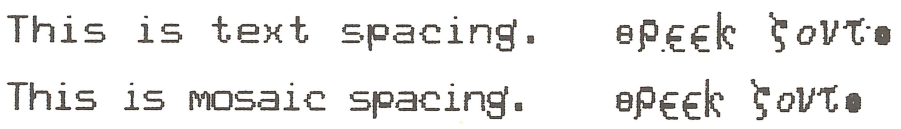


Spacing

Spacing controls the spacing of text characters. The Spacing settings are text and mosaic. Text spacing, which is the default, displays space between each character. Mosaic spacing decreases the space between characters.

Spacing is used with the TEXT function.

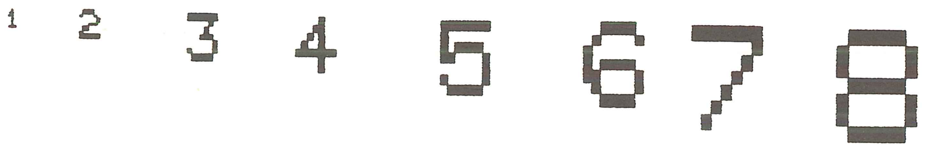
The examples below show mosaic and text spacing used with both ASCII and user-defined characters.



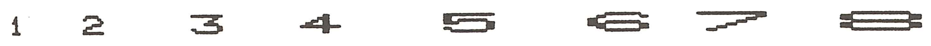
Width

Width specifies the width of text characters. The widths, which are incremented in steps of 1, range from 1 to 8, with 1 being the smallest. The default size is 1. Width is used with the TEXT function.

As the width setting is increased, the Graphics Editor automatically increases the height of the character proportionately. The example below shows the display of each character size if you use the Width key.



To change only the width of the character, but not the height, you must change the setting of the Height key after changing the setting of the Width key. The example below shows the Width settings when the Height setting is maintained at 1.



Write-mode

Write-mode sets the logical operations that control the writing display of the Graphics Editor. The Write-mode settings are replace, erase, complement, and overlay. Replace is the default Write-mode setting.

When using write-mode operations on text, the character cell surrounding the text (and the display of the background under the character cell) is modified, as well as the character itself. Therefore, in some cases it may appear that write-mode settings affect text differently from other objects. This happens because the text is positioned in character cells, which are against the background, while objects are directly on the background. However, the operations on text and other objects are the same.

Replace causes the object or text string to be written on the screen and to replace any other object that has been written in the same location. The new object is written in the color selected with the Color key.

Text used with the replace write-mode setting replaces the object under the entire character cell. The characters are in the writing color and the character cell is in the background color.

Erase inverts the display of the object. The writing color is changed to the background color. This causes the same effect as the use of the Negate key.

When used with text, the character and the character cell are changed to the background color. This has the effect of placing a background-colored box where the text is.

Complement causes the writing color to be complemented. The writing color is changed to the background color. The use of a pattern causes the background-colored part of the pattern to assume the writing color.

A complemented object positioned at the same location as another object does not block out the underlying object. Instead, the complemented object is placed on top of the existing object. The use of a patterned complemented object causes the underlying object to still show wherever the patterned object is not written. Using the complement setting for an object causes any underlying object to be complemented also.

Text written with the complement setting causes the characters to be in the writing color. The complement setting causes the text cell surrounding the character to be transparent; in other words, whatever is under the text cell shows through. Any object underneath the text cell is then complemented.

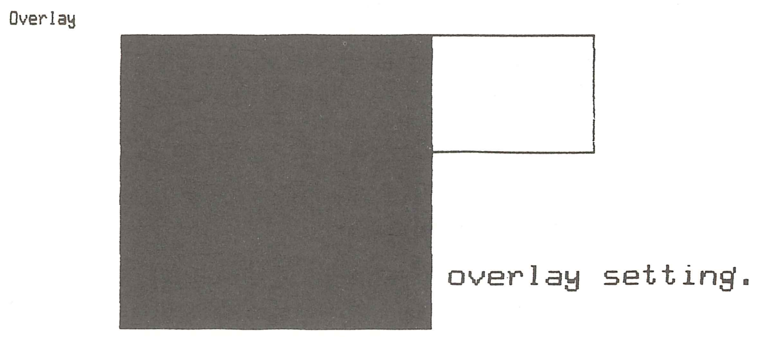
To display text without the text cell being displayed, use the TEXT key and the ENTER key to enter the text. Then use either the BOX key or the LINE key to draw a box around the text. Use the following attribute settings: Shade: on, and Write-mode: complement. Use the ENTER key. The color of the text characters can subsequently be modified by changing the color of the box. The text cell is not displayed.

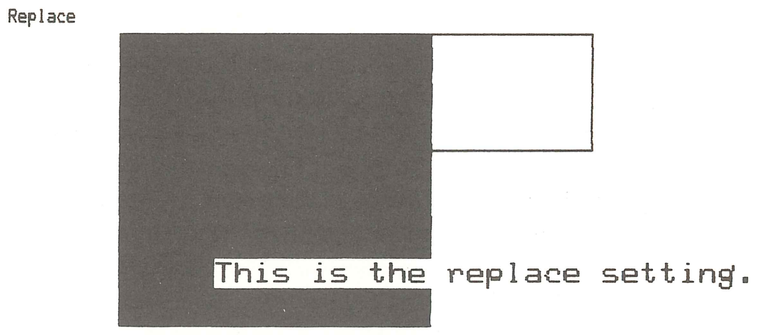
Overlay modifies the display of an object that is positioned at the same location as another object. The overlaid object (the object defined with the overlay setting) is displayed according to the attribute settings that have been specified. The overlaid object affects the existing object only to the extent that it covers the existing object. The overlay setting is the same as laying a cutout on top of another cutout.

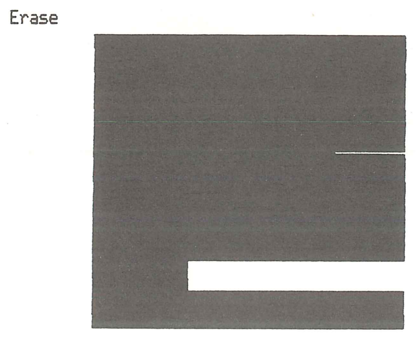
Attribute settings for the overlaid object will also modify the underlying object within the area of the overlaid object.

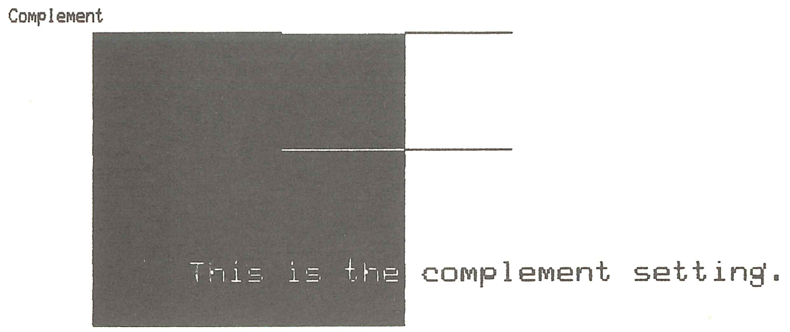
Text written with the overlay setting displays the text cell using the attribute settings of the background object.

The examples below show the Write-mode operations. All four examples start with the large shaded box on the left. The unshaded box on the right and the text string both overlap the shaded box, and are both drawn with the writing mode named on the left side. The text string also indicates the setting for the writing mode.









Selecting Attributes

Attributes can be chosen at either the command level or the object level. The Graphics Editor is at command level when the cursor is blinking next to the command prompt, and is at object level after a function key has been pressed and the graphics cursor is on the screen. If an attribute setting is chosen at the command level, the setting will affect all objects until the attribute setting is changed. If an attribute setting is chosen at the function level, the attribute setting will affect only that type of object. For example, if you choose the color attribute to be red while you are at command level (the graphics cursor is blinking beside the command prompt), everything that is drawn will be red. This includes lines, boxes, circles, curves, and text. However, if you choose the color attribute to be red after you press the BOX key, then only boxes will be red. All boxes will be red until you explicitly change the color attribute, either at the command level or at the function key level.

Attributes can be selected in either of two ways. One method is to successively press the same attribute key to display the consecutive attribute settings. The other method is to use the next or prev key to display the consecutive attribute settings.

For example, if you have pressed the ATTRIBUTE key and the color attribute key, successively pressing the color attribute key will display the names of the available colors. Pressing the ENTER key at a specified color causes that color to be used for drawing.

You can select any attribute for any function. However, only attributes that apply to that function cause any effect. For example, if you select shading to be on while using the TEXT function, the text is not affected. Likewise, if you select shading to be on at command level, all boxes, circles, lines, and curves will be modified (until shading is turned off), but text will not be affected.

6 MODIFYING PICTURES

The process of drawing and saving a picture requires the following steps:

- use of a function key to draw the object
- use of the attribute and interaction keys to modify the object
- use of the ENTER key to save the object in GIGI
- use of the WRITE command to write the file to disk

You can modify the picture at any of these stages. To edit a picture, the picture must be represented in the file copy used by GIGI. You put a file copy into GIGI either by reading a file from disk with the READ command or by saving a picture description with the ENTER command.

This chapter describes the processes, commands, and keys you use to change pictures at any of these stages.

MODIFICATION COMMANDS AND KEYS

Pictures can be modified with both commands and keys. Commands provide functions that change entire objects, for example moving or deleting an object. Keys are used to change aspects of objects, for example the color or the shading or a point on a line.

In addition to the commands that can be used to change objects, other commands simplify the process of editing. These commands, the help commands, do not change the picture; they make it easier to edit, as well as to draw pictures.

Help Commands

The help commands are:

- GROUP
- LABEL
- NAME
- UPDATE
- VERIFY

The GROUP, LABEL, and NAME commands assign labels to ranges of objects and locations in the file. Labels allow you to easily access a specific location or a group of objects. For example, a label of TREE could be assigned to 12 objects that form a tree. To move the tree, you could then specify the label TREE, rather than using the numbers.

The UPDATE command refreshes the screen. When the screen is refreshed, it displays the picture as represented in the GIGI file copy. When editing pictures, it may be necessary to frequently refresh the screen.

The VERIFY command displays a version of the internal GIGI representation of the picture. This display shows the sequential number assigned to each object in the GIGI file copy. This number makes it easy to access a specific object with the typed commands. An example of the VERIFY display is shown below. This was generated from PICTUR.PIC, the picture shown at the end of Chapter 4.

```
Ver T1.05 - Command: v
1:                Text Option (font) = 0
3:                Text Option (width) = 1
5:                Text Option (height) = 2
7:                Text Option (spacing) = 0
9:                Text Option (italic angle) = 0
11:               Text Option (slope angle) = 0
13:               Writing Option (raster op) = 3
15:               Writing Option (negate) = 0
17:               Writing Option (pattern) = 257
19:               Writing Option (pattern multiplier) = 2
21:               Writing Option (shading) = 0
23:               Writing Option (shading y axis) = 0
25:               Writing Option (pixel multiplier) = 1
27:               Writing Option (blink) = 0
29:               Writing Option (color) = 7
31:               Writing Option (background color) = 0
33:               Writing Option (raster op) = 0
35:               Writing Option (pattern) = 255
37:               Writing Option (color) = 4
39:               Writing Option (raster op) = 3
41:               Writing Option (color) = 7
43:               New Page.
44:               Writing Option (raster op) = 0
46:               Writing Option (color) = 4
48:               Text Option (width) = 2
50:               Text Option (height) = 3
52:       1       Text: [20,20] " 11 POINTS".
66:       2       Open Lines: [620,240].
180:              <End of File>
```

The numbers in the first column are internal addresses of objects. The numbers in the second column are the object numbers for each object in the file. Any labels are displayed in the second column. The third column displays a brief description of each attribute setting and object for the given address. The final number in the third column represents the numerical representation that the Graphics Editor uses when drawing pictures.

The object number shown in the second column is the number used when specifying ranges. This example has two objects: object 1 is the text display and object 2 is the rosette.

MODIFYING WHOLE OBJECTS

Commands are used at the command level. You are at command level when the cursor is blinking beside the Command prompt at the top of the screen. The commands you use to modify pictures are:

- COPY
- MOVE
- SCALE
- TILT

The COPY command makes a copy of the specified object or objects. The copy of the object remains at the same location as the original object. This allows you to easily display multiple copies of the same object on the screen.

The MOVE command moves the specified object or objects to the location you indicate.

The SCALE command changes the size of the object or objects you specify.

The TILT command tilts or slants the object or objects you specify. You can indicate the number of degrees the object or objects are tilted.

Modifying Aspects of Objects

To modify an individual object, you first use the ALTER command or the ALTER key. Using the ALTER command allows you to specify a particular object to modify. Using the ALTER key allows you to move to each object in the file. Using either the command or the key puts the Graphics Editor in a state in which you can modify pictures. In the alter state, you can edit one object at a time.

Selecting the Object

If you used the ALTER key to select one object, you will be able to edit only that one object. When you press the ENTER key to save the object, the Graphics Editor exits the alter state and returns to command level.

If you used the ALTER key to select a range, or if you used the ALTER key to enter the alter state, you can move from object to object with the next and the prev keys. The next key moves the graphics cursor to the next object in the file and the prev key moves the graphics cursor to the previous object in the file.

Modifying the Object

After the graphics cursor is positioned on the object to be modified, press the ENTER key. The Graphics Editor then enters a mode in which you can modify the object you selected. At this point you use the same procedures to modify the object as you used to draw it.

Changing Attributes

You use the attribute keys in the same way as if you were drawing a picture. Use the SELECT ATTRIBUTE key to enter a state to change attribute settings. After pressing the SELECT ATTRIBUTE key, you can choose any of the attribute keys and change the settings. Any setting changes are not reflected on the screen display at this point. After changing the setting, press the ENTER key to save the setting. As in drawing pictures, pressing the ENTER key after selecting an attribute does not save the picture.

Using Interaction Keys

You also use the interaction keys in the same way as if you were drawing a picture. For more information on the specific behavior of each attribute key, refer to the material earlier in this chapter and to Chapter 7, Auxiliary and Main Keypad Keys - Reference.

Saving the Changed Picture

After making all the modifications to the picture, use the ENTER key to save the picture. If you used the ALTER command to specify one object, the Graphics Editor then returns to command level. If you specified a range or if you used the ALTER key, the Graphics Editor then prompts you to select the next object. Use either the next or the prev key to select another object, or use the • key to return to command level.

Returning to Command Level

To return to command level without saving the changes you may have made, press the RETURN key on the main keypad. At command level the screen display may not look as you think it should. Use the UPDATE command to refresh the screen. If the display still does not look as it should, you can either modify the picture again or you can start with a new picture. To do this, use the delete key to delete the object and follow the procedures described earlier in the chapter to draw another object.

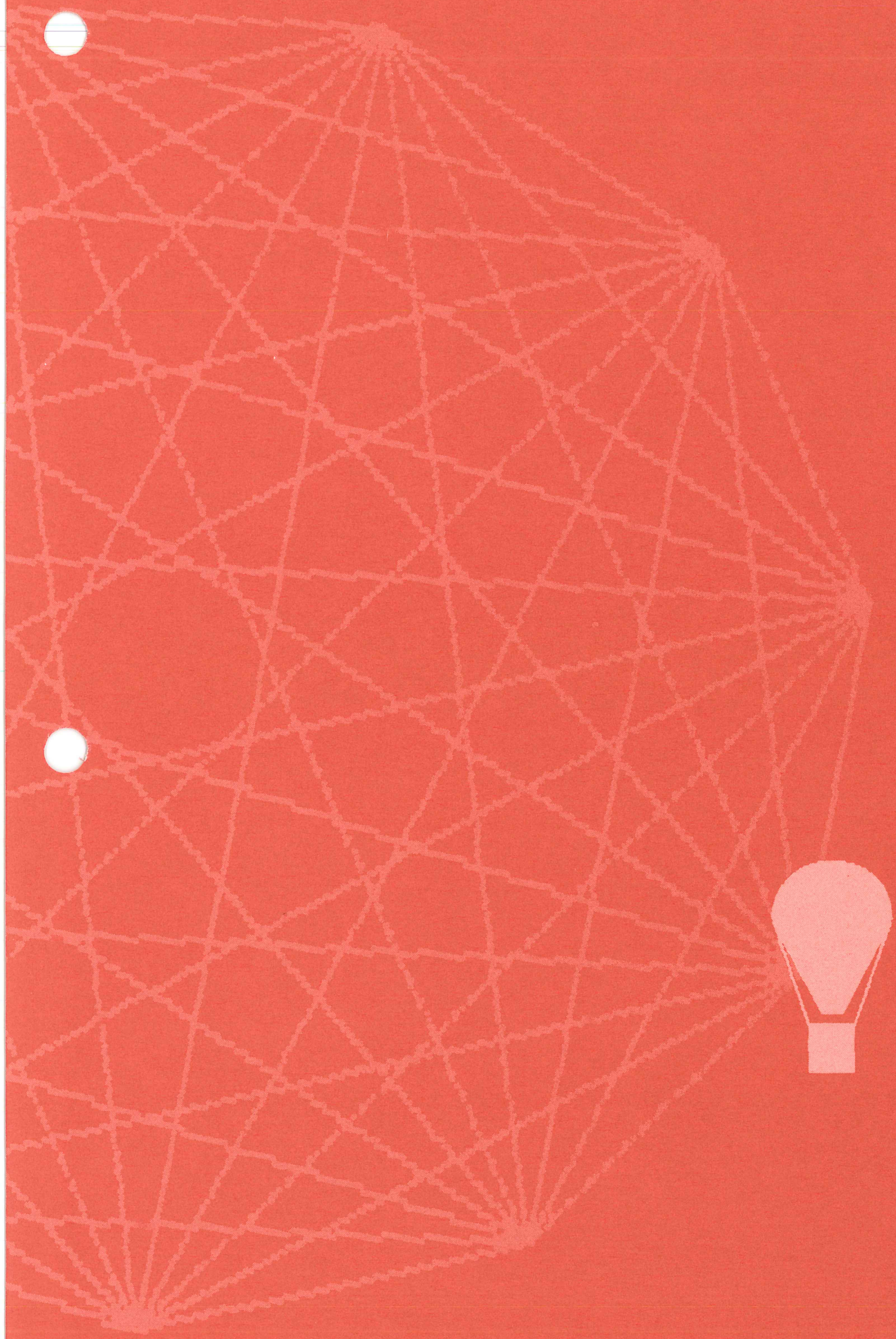
Problems in Editing Pictures

In some cases pictures do not look as you think they should. Often the combination of attribute settings causes unexpected results. When this happens, try to draw objects by changing one attribute at a time. Use the ENTER key to save the picture between each change. The screen display sometimes changes when the picture is written to the internal file copy.

In some cases the location of the object on the screen causes unexpected results. This is because of the way that the ReGIS interpreter in GIGI works. Shading, Pattern, and Pat Mult settings are often affected by this. If you have changed these settings, and the patterns seem to be wrong or in the wrong place, try moving the object up or down with either the arrow or the directional keys. This may correct the problem.

3

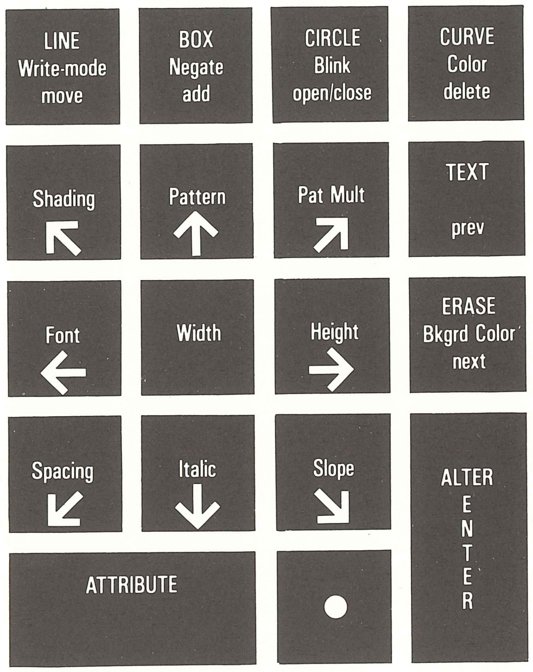
Reference Section



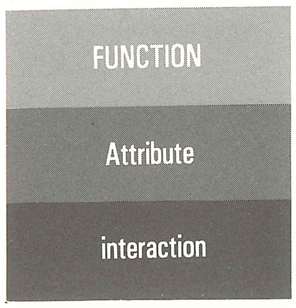
7 AUXILIARY AND MAIN KEYPAD KEYS - REFERENCE

This chapter provides reference information about the auxiliary keypad and the main keypad keys used by the Graphics Editor. It will be helpful if you have already read Section 2, the Usage Section. Section 2 explains the use of the keypads, the steps necessary to draw and edit pictures, and the use of files with the Graphics Editor.

This chapter describes each name shown on the auxiliary keypad overlay as well as each key on the main keypad that has a special Graphics Editor function. A diagram of the auxiliary keypad overlay is shown below.



The three levels on the auxiliary keypad are:



In addition, the auxiliary keypad has control keys which provide control functions.

Each name shown on the keypad is the name of a key. For example, the CURVE key refers to the auxiliary keypad key in the upper right corner. This key can also be called the Color key and the delete key depending on the level used. In some cases,the reference material provides instructions to use the same physical key consecutively, but calls the same physical key by different names. When this happens it is because the key assumes different functions, depending on what key was last used. The use of a key at any level is determined by what sequence of keys was previously used. For example, you may press the TEXT key and then the prev key. Although this is the same physical key, the function it assumes is determined by what was typed previously. The first time you press the key, you will be at the top level and will enable the TEXT function. The second time you press the key, you will use the prev key capabilities.

This chapter lists the keys in alphabetical order. The section for each key contains two parts:

- description
- examples

The description contains the following information: the level at which you use the key, what the key does, and how to use the key. References to other keys are included in this section.

The examples show how the screen display is changed by the use of the key. For most keys, only the part of the screen affected by each step is shown; the example does not show the entire screen. Examples are not provided for keys when the the screen display is not changed or when it is not possible, such as with blinking. The keys are:

- ●
- add
- ALTER
- arrow keys
- ATTRIBUTE
- Bkgrd Color
- Blink
- BOX
- CIRCLE
- Color
- CURVE
- delete
- directional keys
- ENTER
- ERASE
- Font
- Height
- Italics
- LINE
- move
- Negate
- next
- open/close
- Pattern
- Pat Mult
- prev
- Shading
- Slope
- Spacing
- TEXT
- Width
- Write-mode

This chapter uses the following conventions:

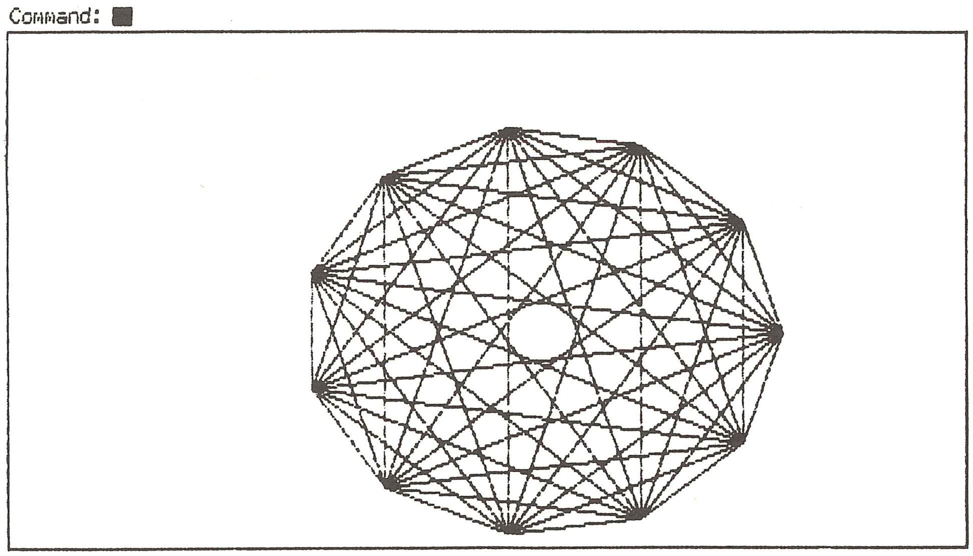
Convention	Description
CAPS	function key, control key
Init Cap	attribute key
lowercase	interaction key
●	current object
1	first object in a picture
n	the relative number of an object in a picture file
,	separator to designate multiple objects
$	the last object in a picture
@	last address range used
*	all objects in a file (1,$)

●

The ● key is a control key.

The ● key returns you to the Graphics Editor command level.

You use the ● key after altering pictures or after using a function key. You must use the ENTER key before using the ● key to save any pictures or changes. Using the ● key without explicitly using the ENTER key causes any changes or drawings just made to be lost.


```
Command: █
```

Add

The add key is an interaction key.

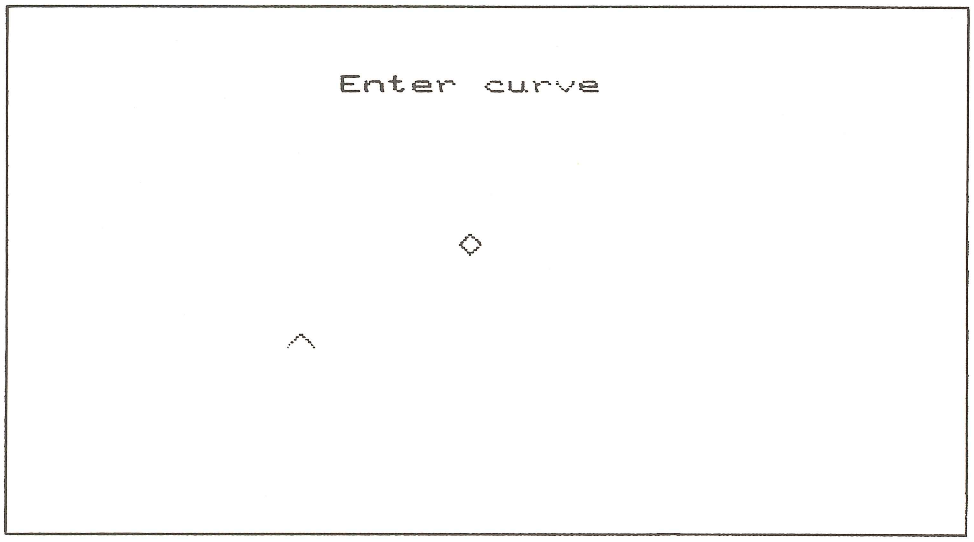
The add key places a mark at the current location. The mark specifies a point that the Graphics Editor uses to draw lines and curves. Before placing the first mark with the add key, you must use either the LINE or the CURVE key.

To modify the placement of marks, you use the move and the delete keys. To position the graphics cursor on marks, you use the prev and the next keys.

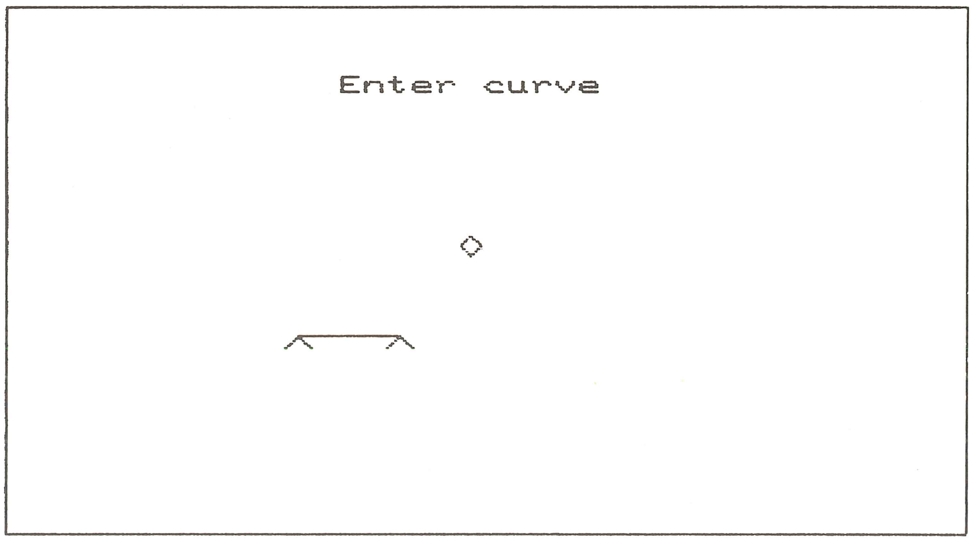
After you use the add key, you can use the directional or arrow keys to move the graphics cursor, the ENTER key to draw the picture, the ATTRIBUTE key to choose an attribute, or other interaction keys.

The mark does not remain visible after you use the ENTER key to save the current picture.

Examples



Enter curve



Enter curve

ALTER

The ALTER key is a control key.

The ALTER key establishes an environment in which you can edit existing pictures and picture files. To use the ALTER key a picture must be available in the GIGI internal file copy.

Using the ALTER key causes the Graphics Editor to display the following prompt:

Select object

where object is selected with the next or prev key. These keys will move the graphics cursor to the next or previous object in the picture.

To return to command level, press the ● key. The ● key does not save any changes made since the use of the ALTER key. Use the ENTER key to save any changes.

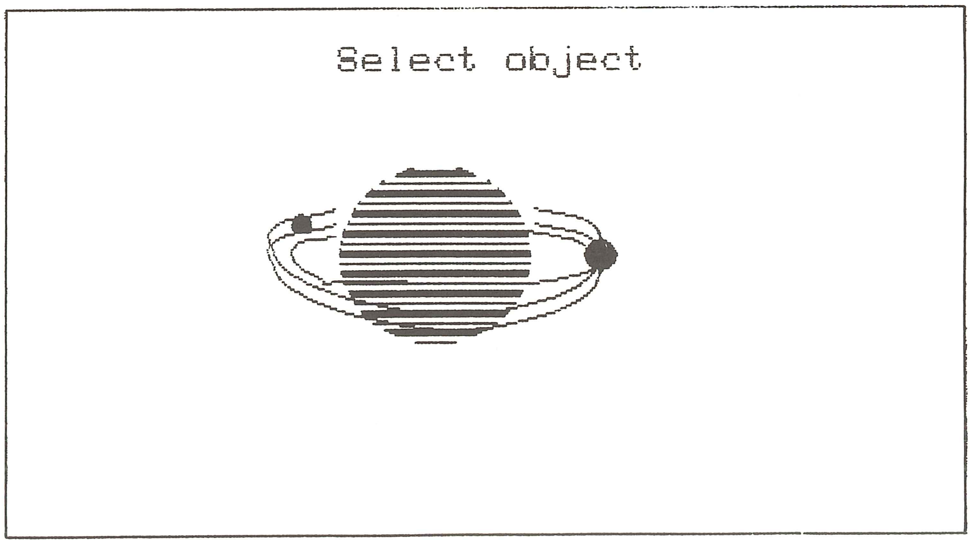
After pressing the ALTER key, position the graphics cursor on the object to be modified using the next or prev keys. After positioning the graphics cursor, press the ENTER key again to select that object. The Graphics Editor displays a prompt that indicates the type of object you are modifying. At this point any of the interaction and attribute keys can be used as if they were being used to draw pictures.

After modifying the picture, use the ENTER key to save the changes in the copy of the file being used by the GIGI. To save the changes on disk, use the WRITE command.

To return to command level, press the ● key. After returning to command level it may be necessary to type the UPDATE command to refresh the screen.

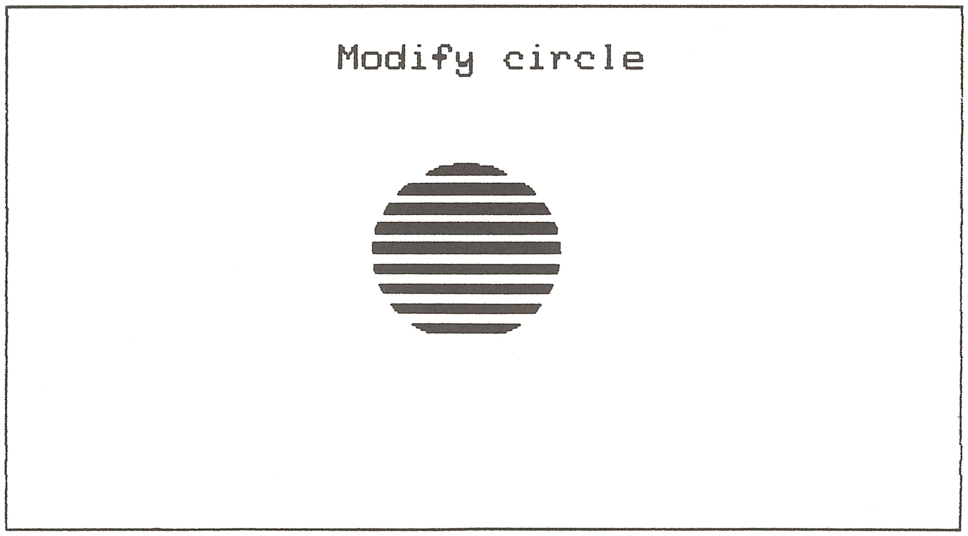
Examples

This is the screen display when the ALTER key is pressed if the picture contains more than one object.

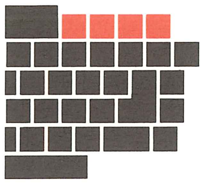

Select object

continued next page

This is the screen display if the picture has only one object. In this example the object is a circle.



arrow keys



The arrow keys are interaction keys and are located on the main keypad.

A single use of an arrow key causes a change of two pixels. Changes are of the following types:

- position of the graphics cursor
- position of the object
- size of the object

The actions of the arrow keys depend on the type of function being drawn or modified and on the last key used. The chart below shows the possible uses of the arrow keys.

Function / Action	Line	Curve	Circle	Box	Text
position the cursor	yes	yes			yes
position the object			yes	yes	yes
change size of object			yes	yes	

Lines and Curves: Positioning the graphics cursor allows you to set the starting point, each subsequent point, and the ending point for lines and curves.

Boxes: Positioning the graphics cursor allows you to place the cursor on any corner of the box, thus enabling other modifications. For more information about modifying boxes, refer to the section on the BOX key.

Positioning the object for boxes means that you can put the box anywhere on the screen. The graphics cursor must be in the upper left or lower right corner to move the box with the arrow keys.

Circles: Positioning the graphics cursor can be done when the Graphics Editor is in a state to create arcs. (Use the next or prev key to position the graphics cursor on the circumference with a radius displayed.) Using the arrow keys then opens and closes the arc and simultaneously moves the graphics cursor around the circumference of the circle.

Positioning the object can be done when the graphics cursor is positioned at the center of the circle. (Use the next or prev keys to position the graphics cursor.) Using an arrow key moves the object in the direction shown on the arrow key.

Changing the size of the object can be done by positioning the graphics cursor on the circumference of the circle without the radius displayed. (Use the next or the prev keys to position the graphics cursor.) The ↑ arrow and the → arrow keys enlarge the circle, and the ↓ arrow and the ← arrow keys shrink the circle.

continued next page

Text: Positioning the graphics cursor for text allows you to specify where the next typed character will be displayed.

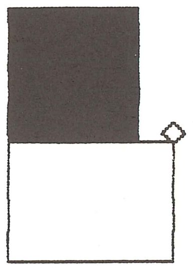
Positioning the object for text allows you to move all text that you have typed since last pressing the TEXT key.

Before using the arrow keys, you must first use a function key. After using an arrow key, you can use attribute keys, the directional keys, and the ENTER key.

The arrow keys perform the same functions as the directional keys on the auxiliary keypad; however, the arrow keys cause movements of two pixels, while the directional keys cause movements of five pixels.

Example This example shows the results of pressing the ⊟ arrow key 25 times after placing a mark.

This example shows a shaded box on top of a non-shaded box. The non-shaded box was originally the same size as the shaded box, but was made wider by pressing the ⊟ arrow key 25 times.



ATTRIBUTE

The ATTRIBUTE key is a control key.

The ATTRIBUTE key enables the selection of attributes. To select an attribute setting, first press the ATTRIBUTE key. The next key used will be on the attribute level. Press the appropriate attribute key. To choose a particular setting for that attribute, continue pressing the attribute key until you see the setting that you want. The screen display does not change yet. To retain that attribute setting, press the ENTER key. To return to command level without changing the current setting for the attribute, press the RETURN key on the main keypad.

You can select attributes at either the command level (when the prompt is beside the command prompt) or the function level (after pressing a function key). In either case the screen display does not reflect the changed attribute setting until the ENTER key is used to save the object. The screen display does not change when you use the ENTER key to set the attribute setting.

Choosing an attribute at command level causes that attribute to affect all drawings until the attribute is changed. Choosing an attribute at function level causes that attribute to affect only objects for that function type.

Choosing an attribute at command level

To choose an attribute at command level, press the ATTRIBUTE key when the graphics cursor is beside the command prompt. Then, press the appropriate attribute key. Choose the setting for the attribute and then press the ENTER key. You are then at command level again.

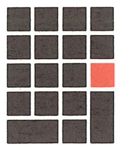
Choosing an attribute at function level

To choose an attribute at function level, press the function key. Press the ATTRIBUTE key and then press the appropriate attribute key. Press the ENTER key to save your attribute choice. (Using the ENTER key when you are selecting an attribute will not save the picture you are drawing.) You can then continue drawing your picture. When you have completed the object, you must then press the ENTER key to save the object.

You can choose the attribute at any time between using the function key and saving the object. Anything already drawn for that object is redrawn to reflect the attribute choice.

To choose more than one attribute, you must repeat the entire process for each attribute. For example, if you want to turn shading on and turn blink on, you must first press the ATTRIBUTE key, press either the Shade key (or the Blink key, it can be done in any order), choose the setting, and then press the ENTER key. You then again press the ATTRIBUTE key, press the Blink key (or the Shade key), choose the setting, and press the ENTER key.

Bkgrd Color



The Bkgrd Color key is an attribute key.

The Bkgrd Color key specifies the color of the screen. To step through the range of colors, press either the RETURN key or the next or prev keys. The screen colors that you can specify are:

- black /0
- blue /1
- red /2
- magenta /3
- green /4
- cyan /5
- yellow /6
- white /7

The numbers beside the color refer to the ReGIS representation of the color. The default screen color is black.

To change the background color, begin by pressing the ATTRIBUTE key, followed by the Bkgrd Color key. Select the color setting by stepping through the colors and using the ENTER key to save the setting. The screen will then be changed to the selected color setting.

To change the background color of a picture you are currently drawing, the picture must first be written to disk with the WRITE command. You can then change the background color, as described above, and read in the file you just wrote to disk. If you change the background color without writing the picture to disk, a new page is created and the picture is not available.

Blink

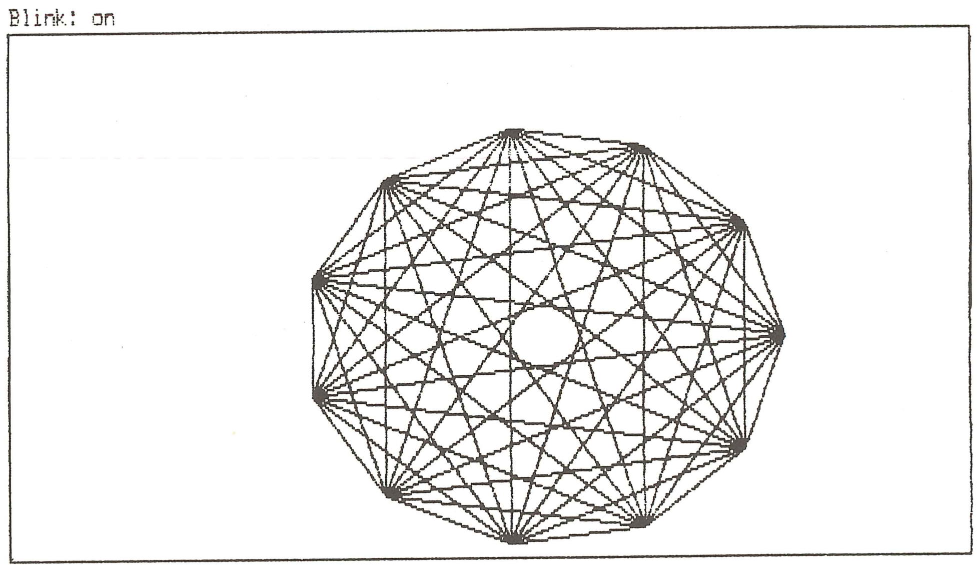
The Blink key is an attribute key.

The Blink key sets blinking to be on or off for the specified object. The default is for blink to be off.

To change the blink setting, you press the ATTRIBUTE key, and then the Blink key. The Graphics Editor then displays the current value for the attribute. To change the value from on to off, or from off to on, you again press the Blink key. After selecting the value, press the ENTER key.

The attribute blink rate differs from the graphics cursor blink rate, so that you will be able to distinguish the blinking graphics cursor when it is on a blinking object.

Example



```
Blink: on
```

BOX

The BOX key is a function key.

The BOX key draws a box. To use the BOX key, the Graphics Editor must be at command level. The top row of keys on the auxiliary keypad, the directional keys, or the arrow keys can be used to position the box and modify its size.

The first method of modifying a box is to use the top row of keys on the auxiliary keypad. The keys have the following effects:

move key	decreases width five pixels
add key	increases width five pixels
open/close key	decreases height five pixels
delete key	increases height five pixels

The second method of modifying boxes is to first use the prev or next keys to position the graphics cursor on one of the corners of the box,and then to modify the size and/or the location of the box with the arrow or the directional keys. The placement of the graphics cursor on a corner determines how the box is modified.

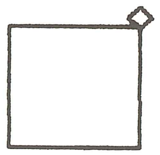
Directional keys move the graphics cursor or the box five pixels, and arrow keys move the graphics cursor or the box 2 pixels in the indicated direction.

Upper left corner and lower right corner: The arrow and directional keys move the box in the direction of the arrow on the overlay or keycap. The size of the box is not changed.

Changes location of box.
Size remains constant.

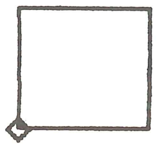
Upper right corner: The arrow and the directional keys move the upper right corner of the box in the direction of the arrow on the keycap. This has the effect of modifying the width and height of the box. The lower left corner of the box remains fixed.

Moves upper right corner.
Left corner remains fixed.



Lower left corner: The arrow and the directional keys move the lower left corner of the box in the direction of the arrow on the keycap. This modifies the width and height of the box. The upper right corner remains fixed.

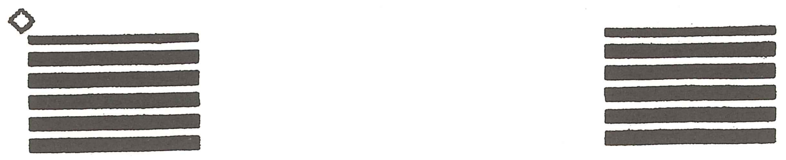
Moves lower left corner of box.
Upper right corner remains fixed.



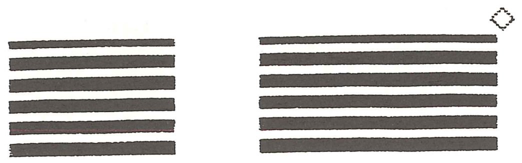
Examples

The example below shows the original box in the left column. The box on the right has been modified with the right directional key. The differences in the box are caused by the cursor position.

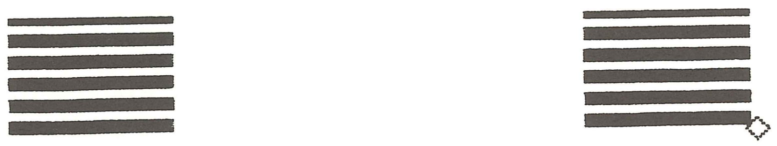
The box is moved to the right.



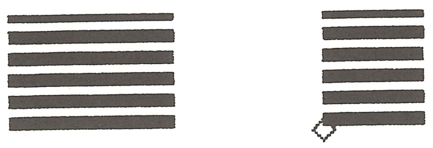
The box is made wider.



The box is moved to the right.



The box is made narrower.



CIRCLE

The CIRCLE key is a function key.

The CIRCLE key draws a circle. You can draw and position circles of differing sizes and create arcs with the CIRCLE key.

The keys on the top row assume special functions after the CIRCLE key is pressed. These keys can modify the circle size and create and modify arcs. You can use these keys to perform all the circle modifications, except positioning the circles. The special functions assumed by the top row of keys are:

move key	decreases radius five pixels
add key	increases radius five pixels
open/close key	decreases arc five pixels
delete key	increases arc five pixels

These keys perform the same functions no matter where the graphics cursor is positioned. Other keys used to modify circles and arcs perform different functions, depending on the position of the graphics cursor.

The graphics cursor can be placed in one of three positions: in the center of the circle (figure 1), on the circumference without a radius displayed (figure 2), and on the circumference with a radius displayed (figure 3). To move the graphics cursor to these various positions, use either the prev or the next key.

Figure 1

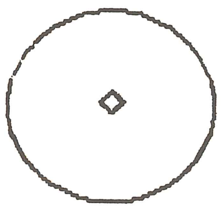


Figure 2

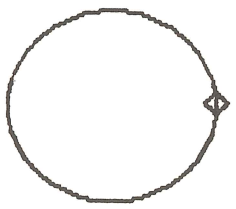
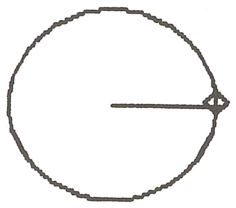


Figure 3



Using the special functions of the top row of keys is the simpler method to use to change the size of circles and arcs.

Changing Circle Location

To change the location of the circle, first position the graphics cursor at the center of the circle (figure 1) by using the next or prev key. Use the directional keys or the arrow keys to move the circle in the direction shown on the keycap or overlay. The directional keys move the graphics cursor five pixels and the arrow keys move it two pixels.

Only the location of the circle changes, the size does not change.

Changing Circle Size

Two methods to change the circle size are available. The first and simplest method is to use the top row of keys: the add key increases the size, and the move key decreases the size of the circle.

The second method is to use the directional and arrow keys. This method also changes the cursor position on the circumference by moving the graphics cursor along the circumference with each key stroke. The graphics cursor must first be positioned on the circumference without a radius displayed (figure 2). The directional and arrow keys can then be used to increase and decrease the size of the circle. The directional keys cause changes of five pixels, and the arrow keys cause changes of two pixels.

The action of the directional keys depends on the location of the graphics cursor on the circumference and the last key used.

This second method also allows you to move the graphics cursor around the circumference of the circle. This is useful when creating arcs at different angles.

Creating and Modifying Arcs

Two methods to create and change arcs are available. The first and simplest is to use the top row of keys: the open/close key decreases the arc and the delete key increases the arc. Successively pressing the open/close key decreases the arc more, and successively pressing the delete key increases the arc more until the circle is completely closed.

The second method is to use the directional and arrow keys. The graphics cursor must first be positioned on the circumference with a radius displayed (figure 3). The directional and arrow keys can then be used to increase and decrease the arc. The directional keys cause changes of five pixels and the arrow keys cause changes of two pixels.

To move the arc to another location, position the graphics cursor at the center of the arc (figure 1) by using the next or prev key. Then use the directional or arrow keys to move the entire object.

You can also specify the tilt of the arc. For example, if you want the arc to open to the left rather than the right, you would use the following sequence of steps.

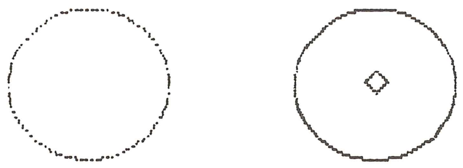
First use the next or prev key to fix the graphics cursor on the circumference without a radius displayed(figure 2). Then use the directional or arrow keys to move the graphics cursor around the circumference. This also changes the size of the circle. (It is still a circle, not yet an arc; but you can correct this after the graphics cursor is positioned where you want it.)When the graphics cursor is positioned on the circumference, use the next or prev keys to position the graphics cursor at the center of the circle. Now use the add or the move keys to increase or decrease the radius of the circle. At this point you can use the open/close key to decrease the arc as much as you want.

continued next page

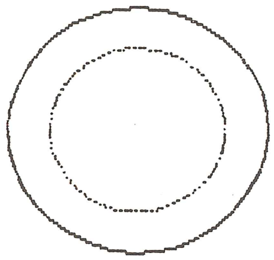
Examples

The examples below use a dotted circle to represent the original circle and a solid circle to represent the modified circle.

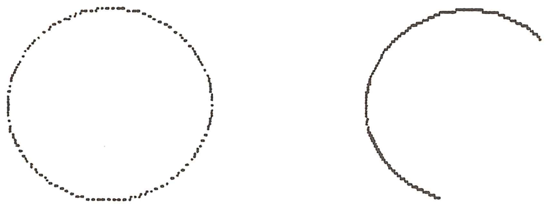
Moving a circle with the right directional key:



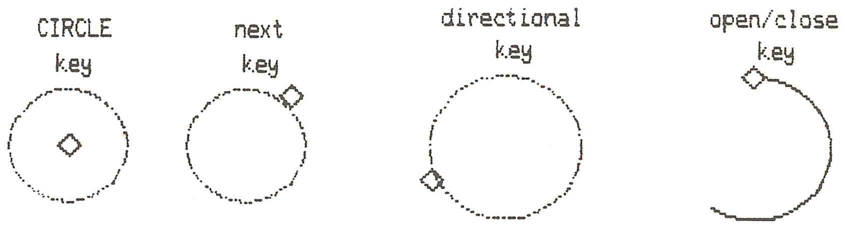
Enlarging a circle with the add key:



Creating a arc with the open/close key:



Tilting the arc:



Color

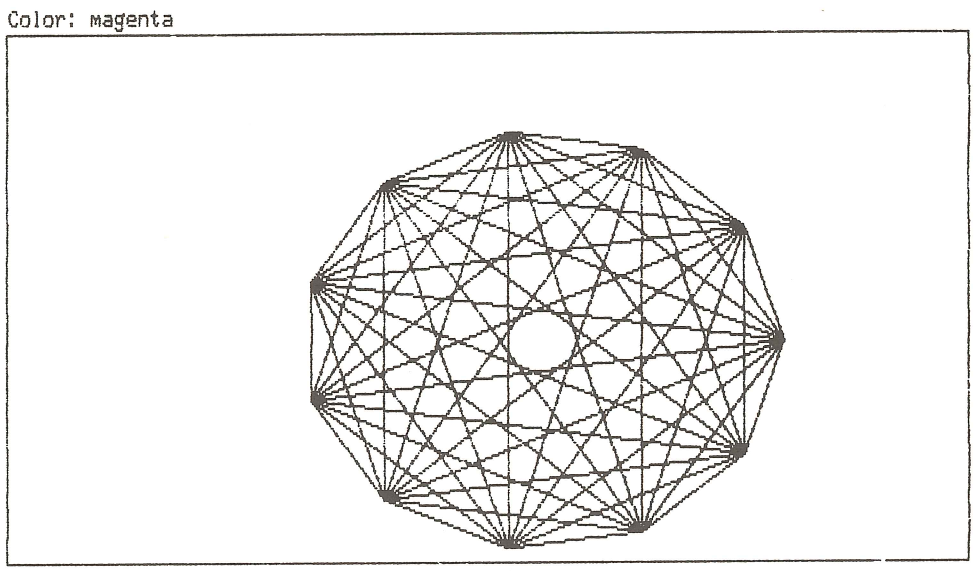
The Color key is an attribute key.

The Color key specifies the writing color. To step through the range of colors, press either the RETURN or the next or the prev keys. The writing colors that you can specify are:

- null
- black /0
- blue /1
- red /2
- magenta /3
- green /4
- cyan /5
- yellow /6
- white /7

The numbers beside the colors indicate the ReGIS representation for the colors. The default writing color is white.

Example



```
Color: magenta
```

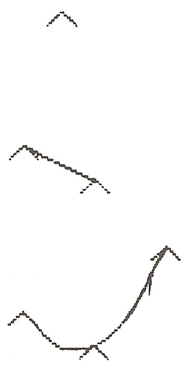
CURVE

The CURVE key is a function key.

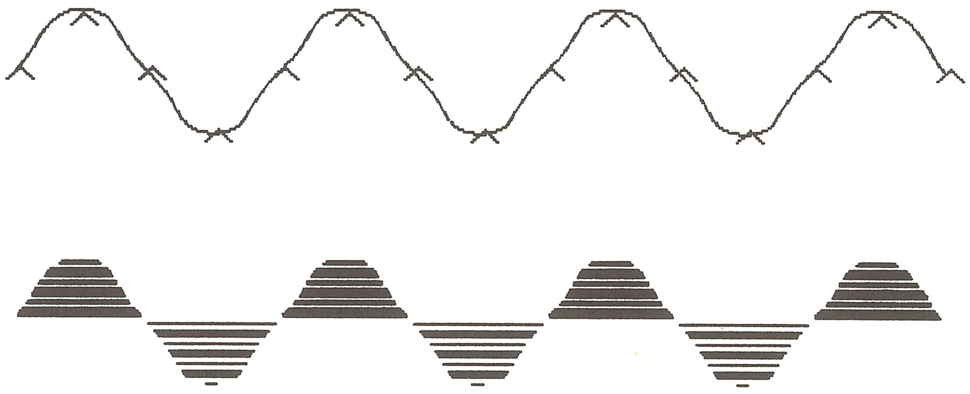
The CURVE key draws a curved line, using marks that you have set as the points for the curve. You set the marks with the add key. For the Graphics Editor to draw a curved line, you must set at least three marks. As you set each mark beyond the second point, the Graphics Editor draws the curve.

Examples

The example below shows first one, then two, and finally three marks placed with the add key when using the CURVE function.



The example below shows first the marks used to draw a sine curve, and second, the curve after being saved with the ENTER key. The curve is highlighted with the Pattern and Shading keys.



delete

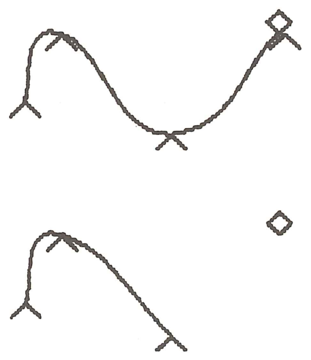
The delete key is an interaction key.

The delete key removes marks and characters. It removes the last mark placed with the add key when drawing lines and curves, and it removes text at the current position when using text.

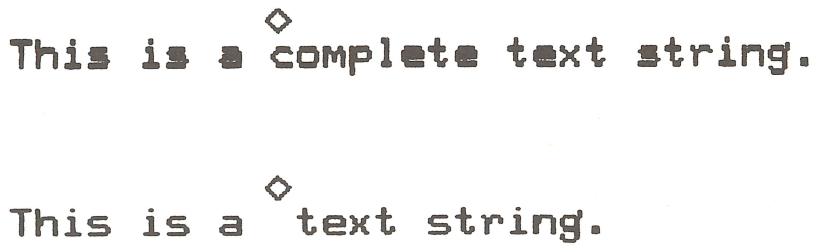
The delete key removes the last mark and leaves the graphics cursor in the current location. The Graphics Editor redraws the line to reflect the removal of the last mark. The delete key removes a mark or character only when the Graphics Editor is at function level.

You can use the ENTER key or any of the attribute, arrow, or directional keys after using the delete key. After removing a mark with the delete key, you store the image by pressing the ENTER key.

The example below shows two curves. The dotted curve has four marks. The solid curve has three marks, the last mark having been deleted with the delete key.



The example below shows two text strings. The first text string indicates the current position by the location of the graphics cursor. The second text string shows the results of using the delete key eight times with the graphics cursor at the same position.



This is a complete text string.

This is a text string.

directional keys

The directional keys are interaction keys and are located on the main keypad.

A single use ofa directional key moves the graphics cursor five pixels. The directional keys do the following:

- position the graphics cursor
- position the object
- change the size of the object

The actions of the directional keys depend on the type of function being drawn or modified and on the last key used. The chart below shows the possible uses of the directional keys.

Function / Action	Line	Curve	Circle	Box	Text
position the cursor	yes	yes			yes
position the object			yes	yes	yes
change size of object			yes	yes	

Lines and Curves

Positioning the graphics cursor allows you to set the starting point, each subsequent point, and the ending point for lines and curves.

Boxes

Positioning the graphics cursor allows you to place the graphics cursor on any corner of the box, thus enabling other modifications. For more information about modifying boxes, refer to the section on the BOX key.

Positioning the object for boxes means that you can put the box anywhere on the screen. The graphics cursor must be on the upper left corner of the box to move the box with the directional keys.

Circles

Positioning the graphics cursor can be done when the Graphics Editor is in a state to create arcs. (Use the next or prev key to position the graphics cursor on the circumference with a radius displayed.) Using the directional keys then opens and closes the arc and simultaneously moves the graphics cursor around the circumference of the circle.

Positioning the object can be done when the graphics cursor is positioned at the center of the circle. (Use the next or prev keys to position the graphics cursor.) Using the directional keys moves the object in the direction shown on the directional key.

Changing the size of the object can be done by positioning the graphics cursor on the circumference of the circle without a radius displayed. (Use the next or the prev keys to position the graphics cursor.) The directional keys enlarge or shrink the circle depending on the location of the graphics cursor on the circumference and on the last directional key used.

Text

Positioning the graphics cursor for text allows you to specify where the next typed character will be.

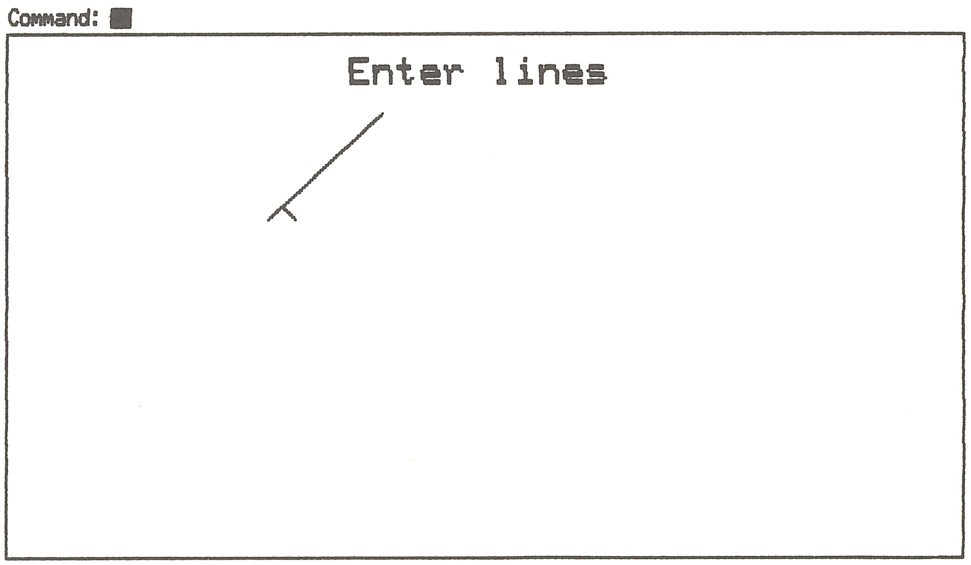
Positioning the object for text allows you to move all text that you have typed since last pressing the TEXT key.

Before using the directional keys, you must first use a function key. After using a directional key, you can use attribute keys, the directional keys, the arrow keys, and the ENTER key.

The directional keys perform the same functions as the arrow keys on the main keypad; however, the directional keys cause movements of five pixels, while the arrow keys cause movements of two pixels.

Examples

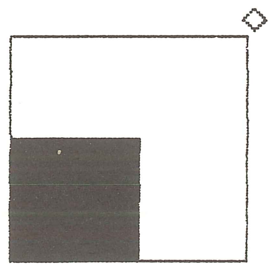
This example shows the results of pressing a ↗ directional key three times after placing a mark while using the LINE function.



Command: ■

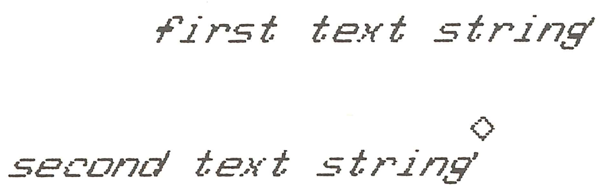
Enter lines

This example shows a shaded box on top of a non-shaded box. The non-shaded box was originally the same size as the shaded box, but was made wider and longer by pressing a diagonal ↗ directional key three times. The graphics cursor is positioned on the upper right corner.



continued next page

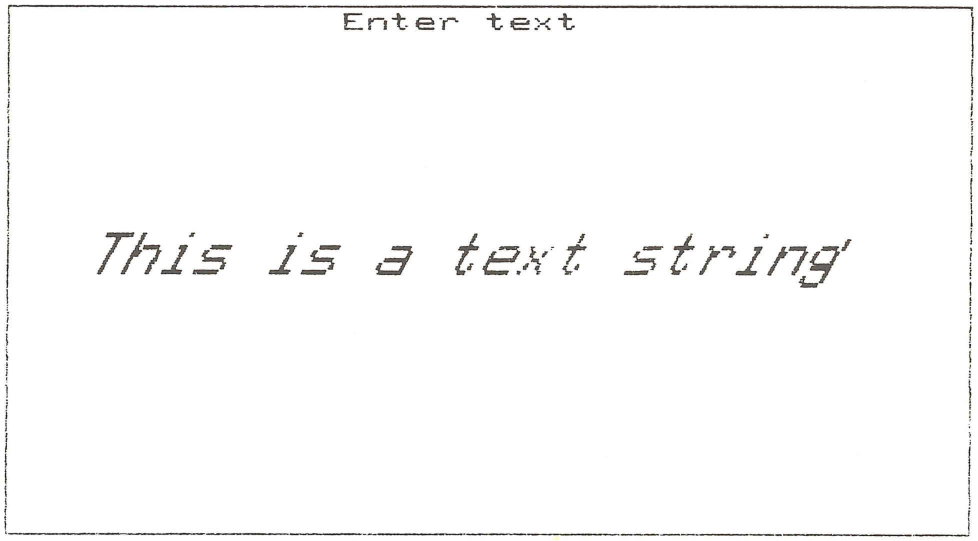
This example shows two text strings. The second text string was positioned in relation to the first text string with a diagonal ↙ directional key.



ENTER

The ENTER key is an interaction key.

The ENTER key writes all changes made to the screen since the last use of a function key. These changes are written to the copy of the file that the Graphics Editor is using. The ENTER key does not write any changes to the file on disk.


Enter text

This is a text string

ERASE

The ERASE key is a function key.

The ERASE key creates a new page in the internal file copy. It does this by clearing the screen display and entering an ERASE command into the internal file copy. It does not change any attribute settings.

The ERASE key can be used at command level or when modifying a picture.

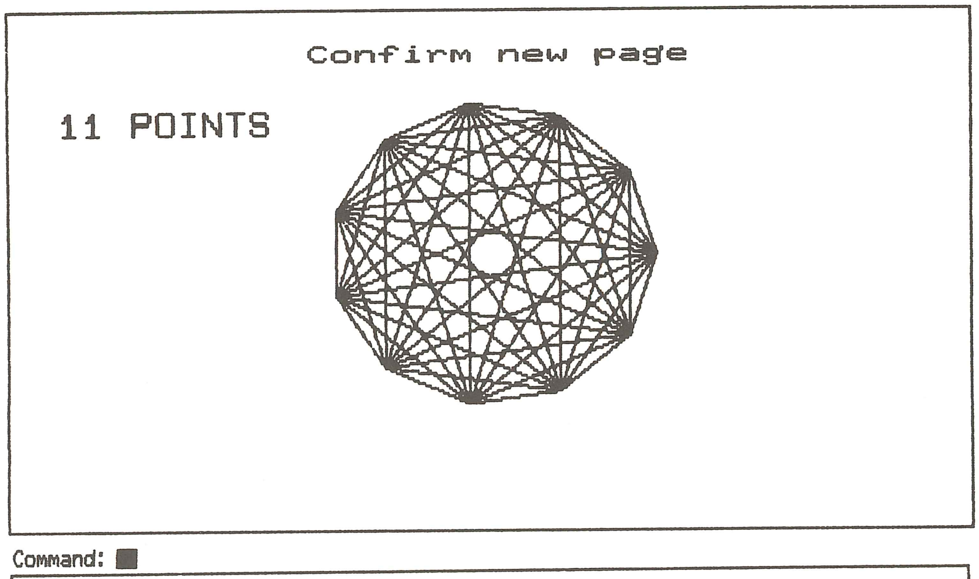
When the ERASE key is used, the Graphics Editor displays a prompt. To erase the screen, press the ENTER key when the prompt is displayed.

Erases or new pages are read into the disk file with the WRITE command. When this disk file is read back into the Graphics Editor, the ERASE command is again read as a new page. Therefore, any objects before the ERASE command are drawn and then deleted when the ERASE command is read. Any objects after the ERASE are displayed.

To delete the erase from the file, use the JOIN command.

Example

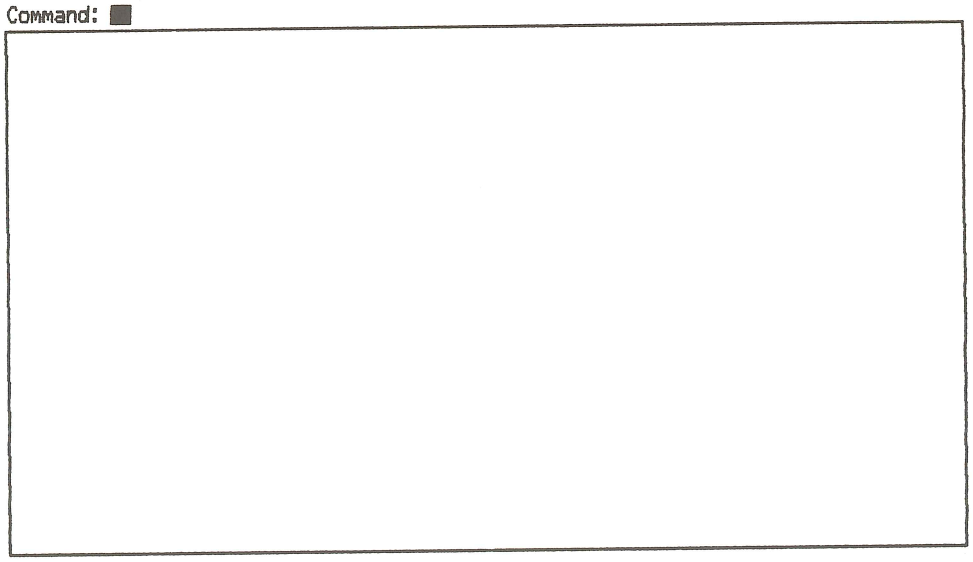
The example below shows two screens. The first screen shows the display after pressing the ERASE key. The second screen shows the display after pressing the ENTER key.



```
Confirm new page

11 POINTS

Command: ■
```



Font

The Font key is an attribute key.

The Font key makes alternate character sets available for use with the TEXT key. Using alternate character sets allows you to use font files created with the Character Set Editor. Characters entered with the TEXT key are then displayed according to the file definitions, rather than as standard ASCII characters.

GIGI has one standard character set and three alternate character sets. The standard character set is the ASCII character set and is always loaded. You can load font files into any or all of the three additional alternate character sets at any one time. The character sets can contain 95 characters. The ASCII character set is the default character set.

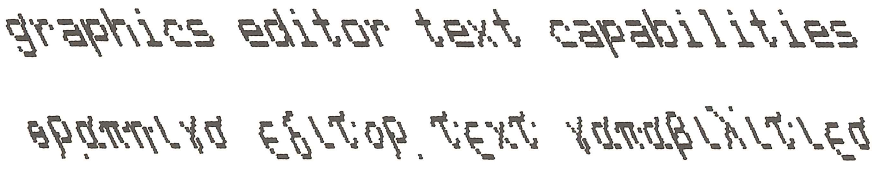
To load a font file into an alternate character set, use the FONT command.

Using the Font key to select an alternate character set with no loaded font file causes characters entered with the TEXT key to display character cells that are either all blank or all shaded.

Font files can be created either with the Character Set Editor or by creating a ReGIS file with a standard text editor.

Example

The example below shows two text strings. The first text string shows a line of text before being saved with the ENTER key and when Font setting is ASCII. The second line of text shows the same line of text after the FONT command has been used to load a Greek font into alternate character set 1 and the Font key has been used to select alternate character set 1.



Height

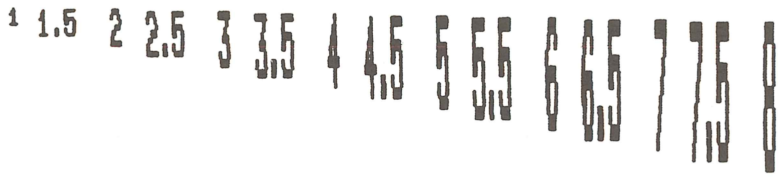
The Height key is an attribute key.

The Height key sets the height that will be used by the Graphics Editor for text characters.The width and spacing of characters are not changed unless you explicitly change them with the Width or the Spacing keys.

The height sizes, which are incremented in steps of .5, range from .5 to 8. Size 1 is the default height size.

Examples

The displays in the example below show the comparative values of the heights. These sizes may appear as different sizes on different monitors.



Italics

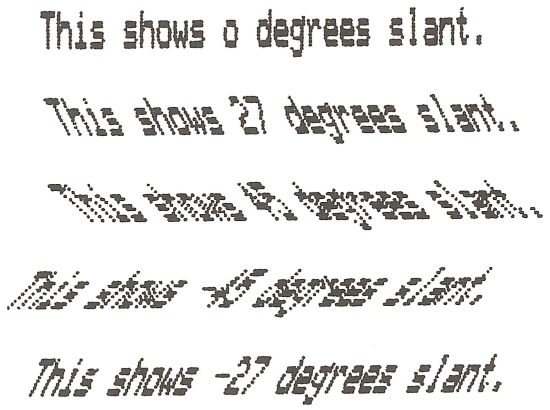
The Italics key is an attribute key.

The Italics key slants characters.These characters are then considered to be italics. The Graphics Editor slants characters in four ways:

- +27 degrees
- +45 degrees
- –27 degrees
- –45 degrees

Using both the Height key to increase character height and the Italics key may cause characters in a string to overlap, if the characters are adjacent.

Examples The example below shows the italics settings.



```
This shows 0 degrees slant.
This shows +27 degrees slant..
This shows +45 degrees slant...
This shows -45 degrees slant.
This shows -27 degrees slant.
```

LINE

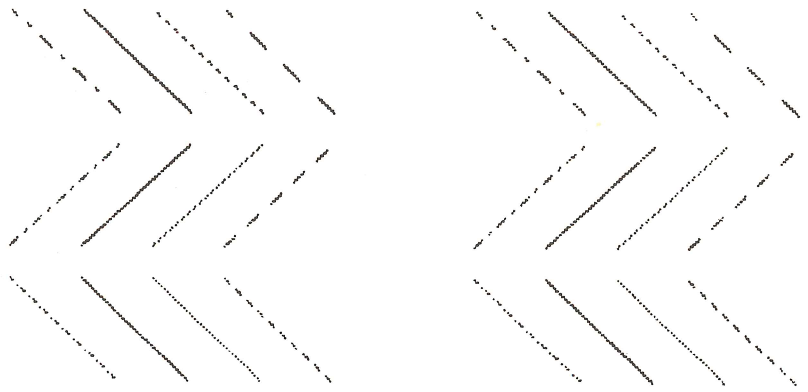
The LINE key is a function key.

The LINE key draws a line connecting the marks you specify with the add key. You must specify at least two marks before the Graphics Editor draws a line.

After you use the LINE key, you can use the attribute keys, the interaction keys, and the ENTER key.

Examples

The examples below show lines created with the LINE function. Each unattached line is an object. Different Pattern and Pat Mult settings have been used to vary the appearances of the lines.



move

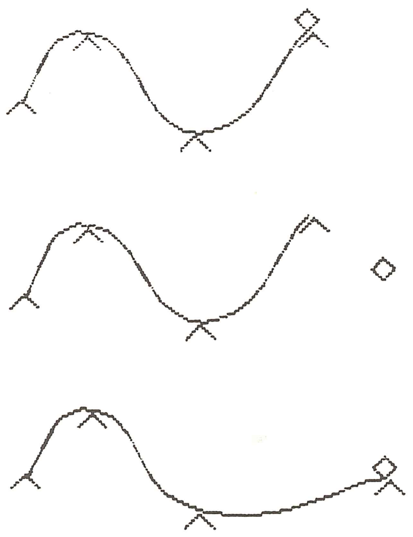
The move key is an interaction key.

While using the LINE and CURVE functions, the move key moves marks. The move key removes the last mark placed after using the ENTER key, and places the mark at the current graphics cursor position. While using the CIRCLE and BOX functions, the move key decreases the size of the object.

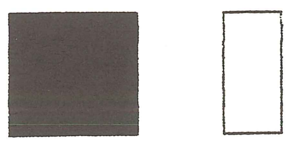
You can use the move key after using the LINE, CURVE, CIRCLE, and BOX keys. After using the move key, you can use the attribute keys, the other interaction keys, and the ENTER key.

Examples

The example below shows three figures. In the first figures, the curve has four marks with the graphics cursor positioned on the last mark. The second figure shows the curve after the graphics cursor was moved to another location with a directional key. The third curve shows the redrawn curve after pressing the move key. The last mark (on which the graphics cursor was previously positioned) has been moved to the new location of the graphics cursor. The curve is redrawn to reflect this.

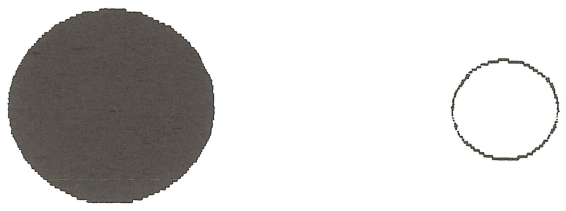


The example below shows a non-shaded box beside a shaded box. The shaded box was modified with the move key to produce a box the size of the non-shaded box.



continued next page

The example below shows a non-shaded circle beside a shaded circle. The shaded circle was modified with the move key to produce a circle the size of the non-shaded circle.



Negate

The Negate key is an attribute key.

The Negate key inverts the display of the object. The writing color is changed to the background color, and the character cell other than the letter is changed to the writing color.

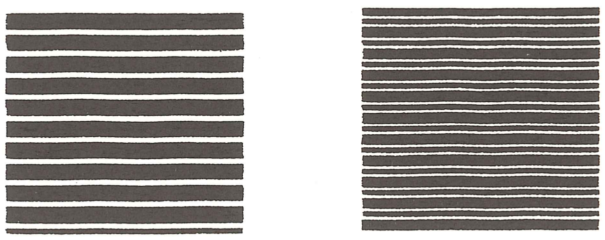
Examples

The example below shows text characters written with the Negate setting on.

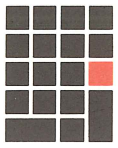
This is an example

of the negate attribute.

The example below shows two boxes. The box on the left was written with Shading on, and the dot-dashed Pat Mult setting. The box on the right was written with the same attributes, and also with the Negate setting on.



Next



The next key is an interaction key.

The next key moves the graphics cursor to the next position. The next position can be any of the following, depending on the last key you have used:

- position on a circle or box (BOX, CIRCLE keys)
- the next mark (LINE, CURVE keys)
- the next object (ALTER key)
- the next character (TEXT key)

The next key moves the graphics cursor to different positions on the object if the last key used was either the CIRCLE or BOX key. Changing the graphics cursor position on circles and boxes allows you to modify the position and shape of the object. For more information about modifying boxes and circles, refer to the sections on boxes and circles.

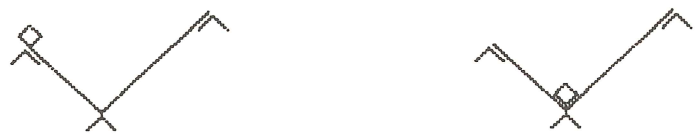
The next key moves the graphics cursor to the next mark if the last key used was either the LINE or CURVE key. For more information about using marks, refer to the sections on the move, add, and delete keys.

The next key moves the graphics cursor to the next object if the last key used was the ALTER key. You can move the graphics cursor to each successive object by using the next key consecutively. For more information about modifying objects after moving the graphics cursor, refer to the section on the ALTER key.

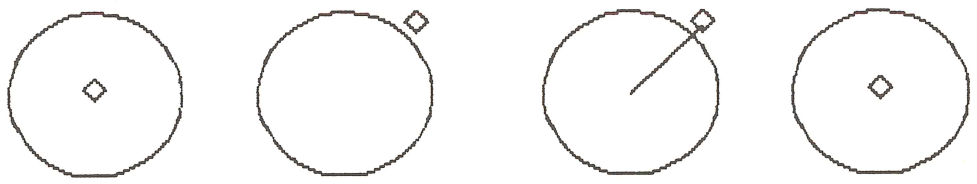
The next key moves the graphics cursor to the next character in a text string if the last key used was the TEXT key. You can move the graphics cursor to each successive character by using the next key consecutively. For more information about the TEXT key, refer to the section on the TEXT key.

Examples

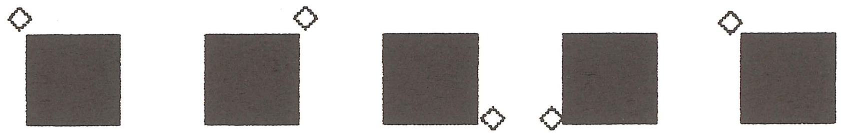
The example below shows two copies of the same line. In the second line the graphics cursor has been moved from one mark to the next mark with the next key.



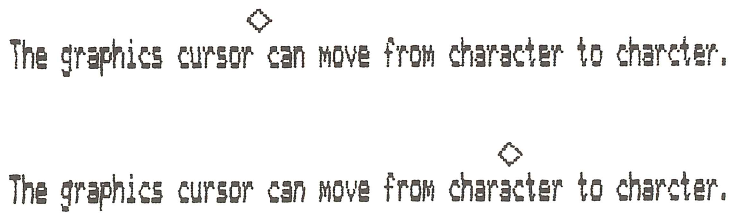
The example below shows four copies of the same circle. In each circle the next key has been used to move the graphics cursor through the three cursor positions for circles. In the final circle, the graphics cursor is back in the original position.



The example below shows five copies of the same box. In each box the next key has been used to move the graphics cursor through the four cursor positions for boxes. In the final box, the graphics cursor is back in the original position.



The example below shows two copies of the same text string. In the second text string, the graphics cursor position has been moved with the next key.

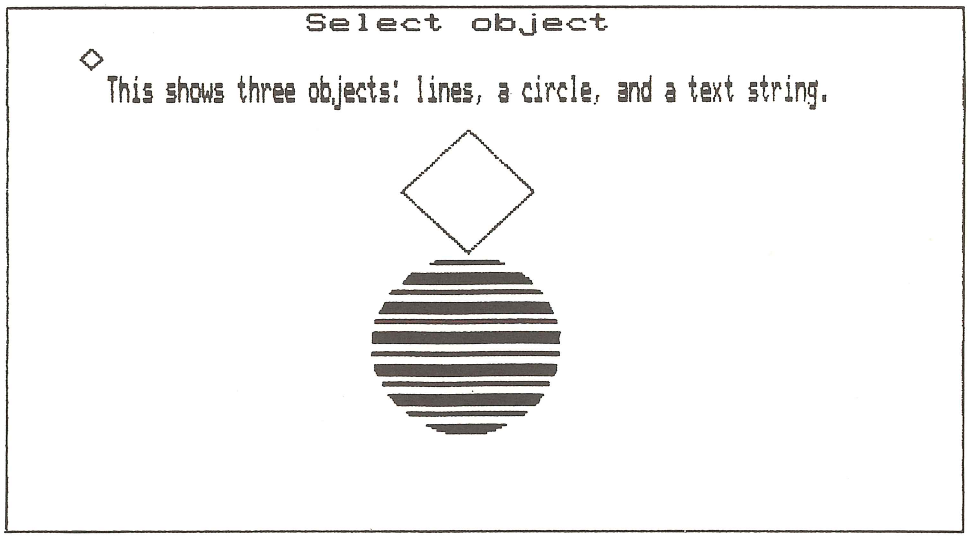


The graphics cursor can move from character to charcter.

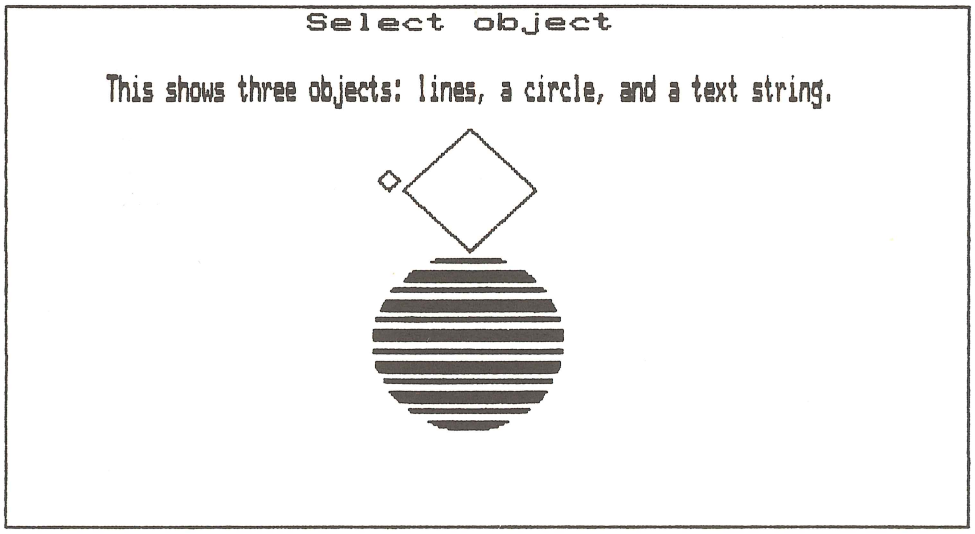
The graphics cursor can move from character to charcter.

continued next page

The example below shows two copies of the same screen display after pressing the ALTER key. In the second screen display, the graphics cursor position has been moved from one object to the next object with the next key.


Select object

This shows three objects: lines, a circle, and a text string.


Select object

This shows three objects: lines, a circle, and a text string.

open/close

The open/close key is an interaction key.

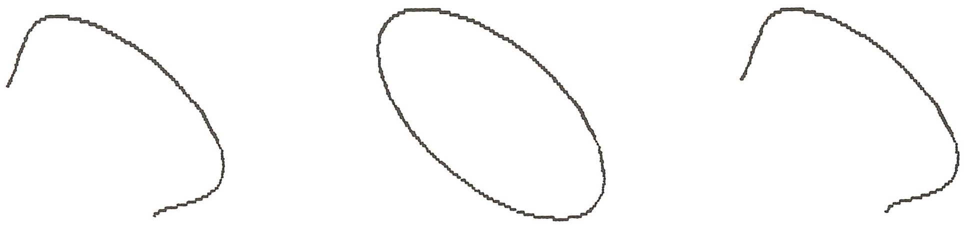
The open/close key opens and closes lines and curves drawn with the LINE and CURVE keys, and modifies arcs drawn with the CIRCLE key.

When used with the LINE and CURVE keys, the open/close key causes a line to be drawn or erased, depending on the line's current state, between the first and the last points of a line or curve. If the line or curve is open, the open/close key closes it. If the line or curve is closed, the open/close key opens it.

When used with the CIRCLE key, the open/close key modifies the arc by five pixels.

Example

The example below shows three copies of the same curve. The first curve was drawn with the CURVE function. The second curve was created with the open/close key. The third curve was created from the second curve by using the open/close key again.



Pattern

The Pattern key is an attribute key.

The Pattern key controls the writing pattern used by the Graphics Editor. You can set the following patterns:

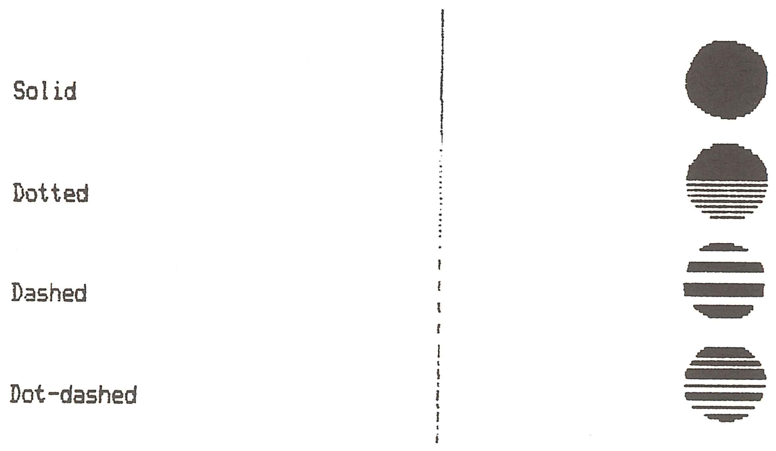
- solid
- dotted
- dashed
- dot-dashed

The default pattern is solid.

The dotted, dashed, and dot-dashed patterns draw the specified type of line.

The Pattern key controls writing and shading patterns. Using the Pattern key with shading turned off causes the outline of the specified object to be drawn with the specified pattern. Using the Pattern key with shading turned on causes the object to be shaded with the specified pattern.

Examples Below are samples of lines and circles drawn with the four writing patterns.



Pat Mult

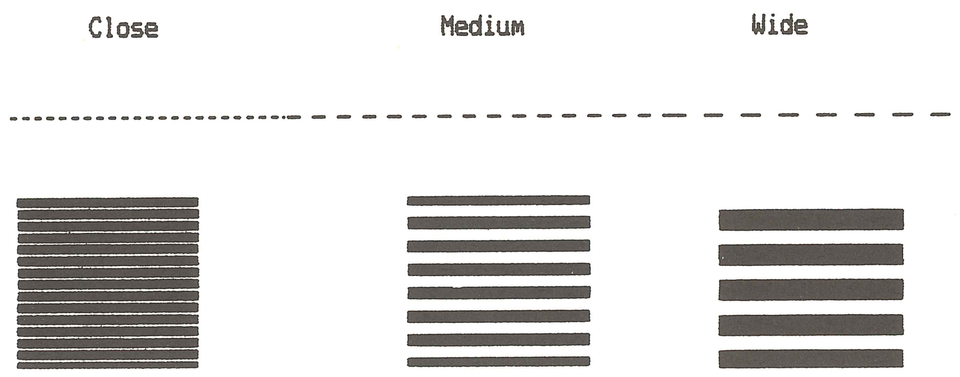
The Pat Mult (Pattern Multiply) key is an attribute key.

The Pat Mult key multiplies the length of the pattern selected with the Pattern key. You can select one of three pattern multipliers:

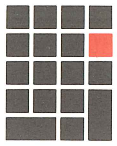
- close
- medium
- wide

The Pat Mult key modifies the patterns affected by the Pattern key: dotted, dashed, and dot-dashed. The Pat Mult key can be used to modify the line-drawing patterns and the shading patterns for lines, boxes, circles, and curves.

Examples Below are samples of lines and circles drawn with the three Pat Mult settings.



prev



The prev key is an interaction key.

The prev key moves the graphics cursor to the previous position. The previous position can be any of the following, depending on the last key you have used:

- position on a circle or box (BOX, CIRCLE keys)
- the previous mark (LINE, CURVE keys)
- the previous object (ALTER key)
- the previous character (TEXT key)

The prev key moves the graphics cursor to different positions on the object if the last key used was either the CIRCLE or BOX key. Changing the graphics cursor position on circles and boxes allows you to modify the position and size of the object. For more information about modifying boxes and circles, refer to the sections on boxes and circles.

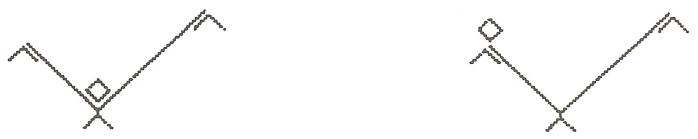
The prev key moves the graphics cursor to the previous mark if the last key used was either the LINE or CURVE key. For more information about using marks, refer to the sections on the move, add, and delete keys.

The prev key moves the graphics cursor to the previous object if the last key used was the ALTER key. You can move the graphics cursor to each successive object by using the prev key consecutively. For more information about modifying the objects after moving the graphics cursor, refer to the section on the ALTER key.

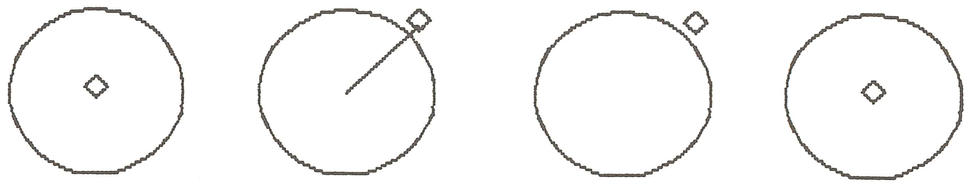
The prev key moves the graphics cursor to the previous character if the last key used was the TEXT key. You can move the graphics cursor to each successive character by using the prev key consecutively. For more information about the TEXT key, refer to the section on the TEXT key.

Examples

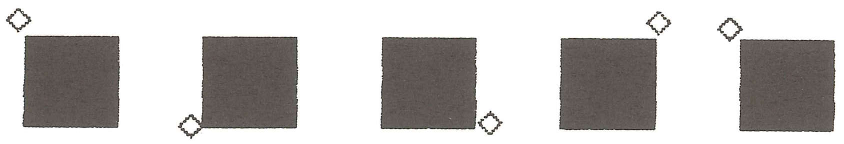
The example below shows two copies of the same line. In the second line the graphics cursor has been moved from one mark to the next mark with the prev key.



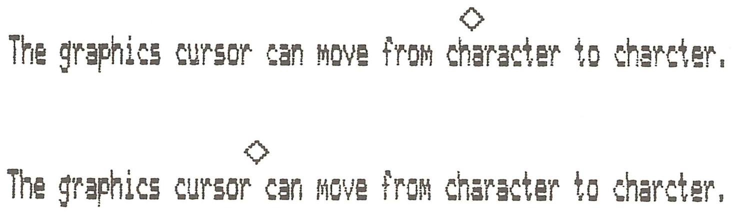
The example below shows four copies of the same circle. In each circle the prev key has been used to move the graphics cursor through the three cursor positions for circles. In the final circle, the graphics cursor is back in the original position.



The example below shows five copies of the same box. In each box the prev key has been used to move the graphics cursor through the four cursor positions for boxes. In the final box, the graphics cursor is back in the original position.



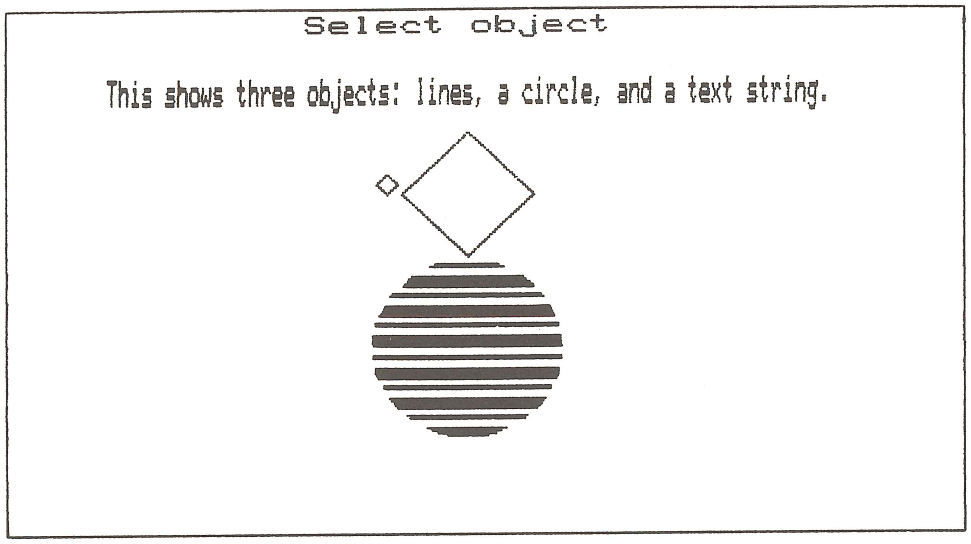
The example below shows two copies of the same text string. In the second text string, the graphics cursor position has been moved with the prev key.


The graphics cursor can move from character to charcter.

The graphics cursor can move from character to charcter.

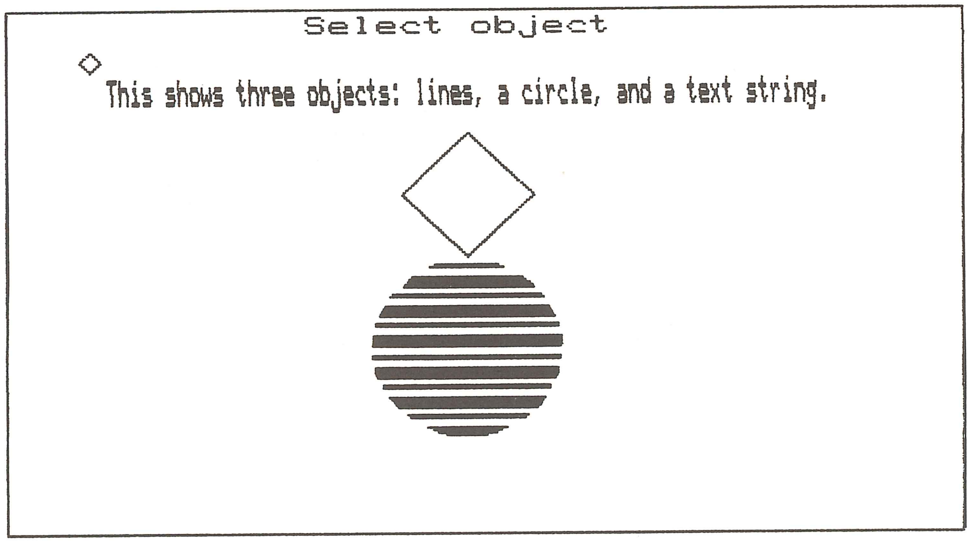
continued next page

The example below shows two copies of the same screen display after pressing the ALTER key. In the second screen display, the graphics cursor position has been moved from one object to the previous object with the prev key.



Select object

This shows three objects: lines, a circle, and a text string.



Select object

This shows three objects: lines, a circle, and a text string.

Shading

The Shading key is an attribute key.

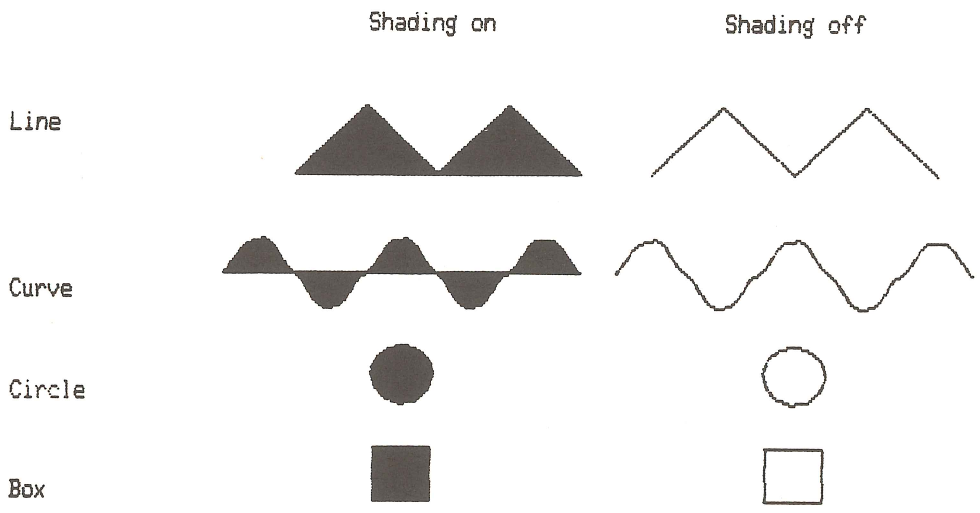
The Shading key shades objects. The color that the object is shaded is set with the Color key. The Shading key is used to turn shading on and off; the default value is off.

You can use the Shading key with the LINE, BOX, CIRCLE, and CURVE keys.

After using the Shading key, you can use other attribute keys, the ENTER key, or the interaction keys.

Examples

The examples below show a line, a curve, a circle, and a box with shading on and with shading off.



Slope

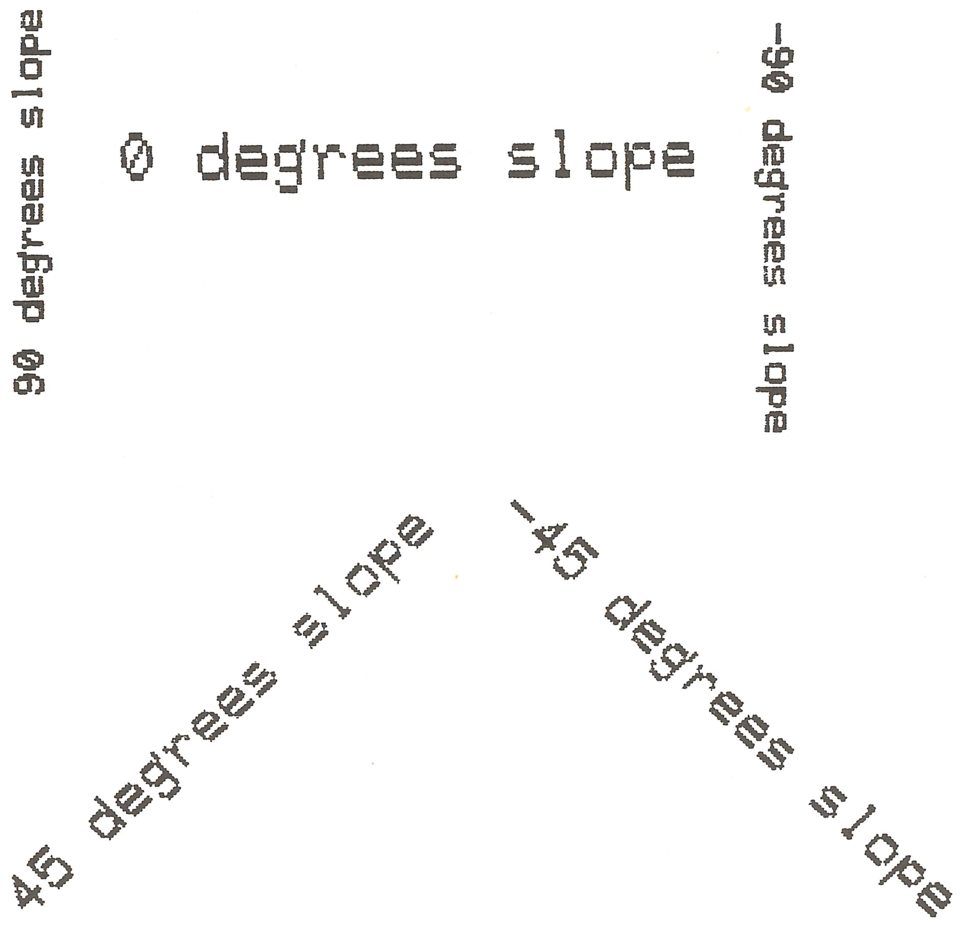
The Slope key is an attribute key.

The Slope key controls the slant at which text is written. The Slope settings are:

- 0 degrees
- 45 degrees
- 90 degrees
- –45 degrees
- –90 degrees

The default is 0 degrees.

Examples The example below shows the slope options.



Spacing

The Spacing key is an attribute key.

The Spacing key controls the spacing of text characters. Two settings are available:

- text
- mosaic

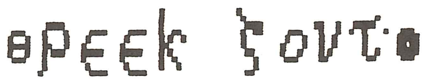
The default value is text.

Text spacing displays text using space between each typed character. Each character is a matrix of dots eight pixels wide by ten pixels high.

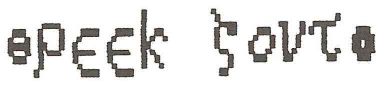
Mosaic spacing decreases the spacing between adjoining characters. If user-defined character sets are used, space between adjoining characters is removed so that the characters appear attached. If the default ASCII character set is used, spacing is decreased between adjoining characters so that the characters are closer together.

Examples The examples below show both text and mosaic spacing for both ASCII and user-defined characters.

```
This is text spacing.
```



```
This is mosaic spacing.
```



TEXT

The TEXT key is a function key.

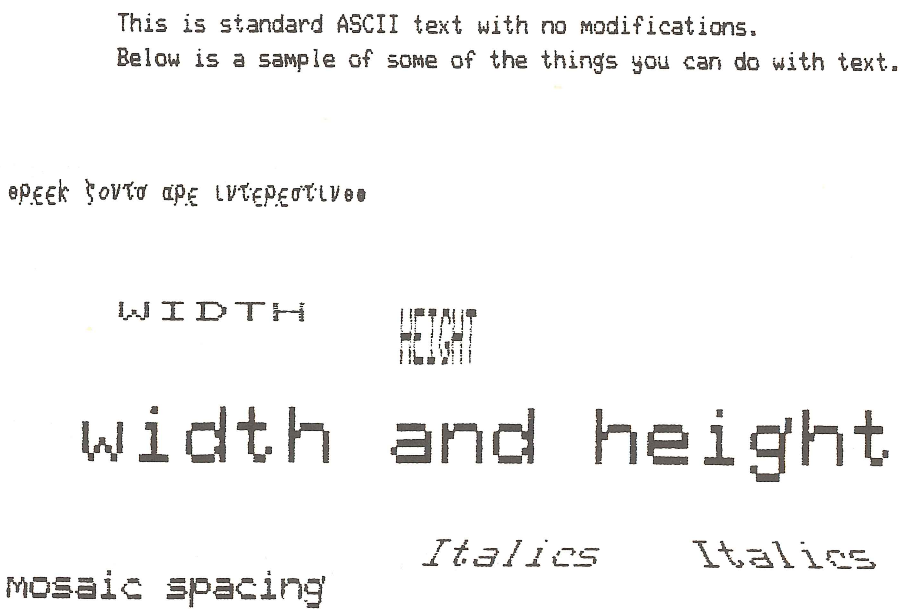
The TEXT key allows you to enter text characters.

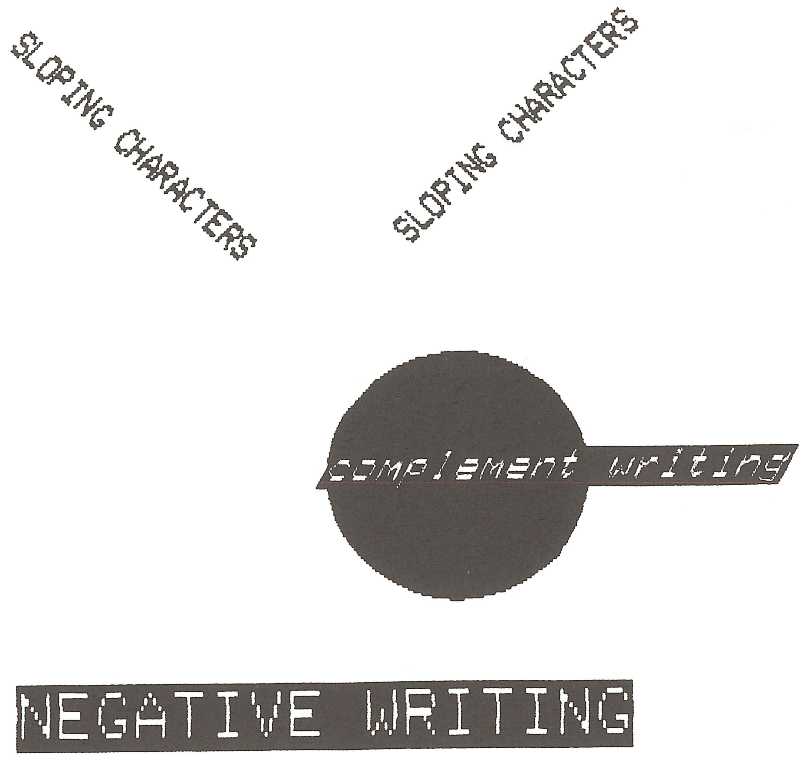
With the Font key, you can select alternate character sets to be used with the TEXT key. Refer to the sections on the Font key and the LOAD command for information about using alternate character sets.

The attribute keys that affect text displays are: Write-mode, Negate, Blink, Color, Font, Width, Height, Background Color, Spacing, Italics, and Slope.

After using the TEXT key, you can use the main keypad keys to type text, as well as the attribute keys, interaction keys, and the ENTER key.

Examples The examples below show text created with various attributes, including Font, Width, Height, Spacing, Italic, and Slope.


```
This is standard ASCII text with no modifications.
Below is a sample of some of the things you can do with text.
```


SLOPING CHARACTERS
SLOPING CHARACTERS
complement writing
NEGATIVE WRITING

Width

The Width key is an attribute key.

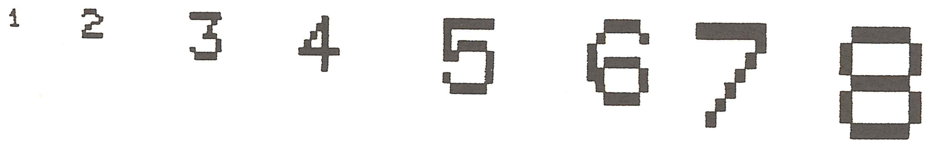
The Width key controls the width of text characters. The height or spacing is not affected. The width settings range from 0 to 5 with size 0 being the smallest and size 5 being the largest. Each larger width setting increases the width of the characters one and one-half times, except for size 0 which is half the width of size 1. Size 1 is the default size.

The height of the characters is changed to correspond to the width changes. To change only the width without changing the height, use the Height key after using the Width key to change the height.

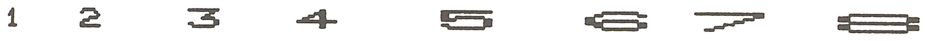
To select a width, you first press the ATTRIBUTE key. You then press the Width key. You continue to press the Width key until the setting you want to use is displayed. You then press the ENTER key to retain your choice.

Examples

The example below shows the width sizes using the default height sizes.



The displays in the example below show the comparative values of the widths.



Write-mode

The Write-mode key is an attribute key.

The Write-mode key sets the logical operations that control the writing display of the Graphics Editor. The write-mode settings are:

- replace
- erase
- complement
- overlay

Overlay is the default Write-mode operation.

When using write-mode operations on text, the character cell surrounding the text (and the display of the background under the character cell) is modified, as well as the character itself. Therefore, in some cases it may appear that write-mode settings affect text differently from other objects. This happens because the text is positioned in character cells, which are against the background, while objects are directly on the background. However, the operations on text and other objects are the same.

Replace: causes the object or text string to be written on the screen and to replace any other object that has been written in the same location. The new object is written in the color selected with the Color key.

Erase: inverts the display of the object. The writing color is changed to the background color. This causes the same effect as the use of the Negate key.

When used with text, the character and the character cell are changed to the background color. This has the effect of placing a background-colored box where the text is.

Complement: causes the writing color to be complemented. The writing color is changed to the background color. The use of a pattern causes the background-colored part of the pattern to assume the writing color.

A complemented object positioned at the same location as another object does not block out the underlying object. Instead, the complemented object is placed on top of the existing object. The use of a patterned complemented object causes the underlying object to still show wherever the patterned object is not written. Using the complement setting for an object causes any underlying object to be complemented also.

Text written with the complement setting causes the characters to be in the writing color. The complement setting causes the text cell surrounding the character to be transparent; in other words, whatever is under the text cell shows through. Any object underneath the text cell is then complemented.

To display text without the text cell being displayed, use the TEXT key and the ENTER key to enter the text. Then use either the BOX key or the LINE key to draw a box around the text. Use the following attribute settings: Shade: on, and Write-mode: complement. Use the ENTER key. The color of the text characters can subsequently be modified by changing the color of the box. The text cell is not displayed.

continued next page

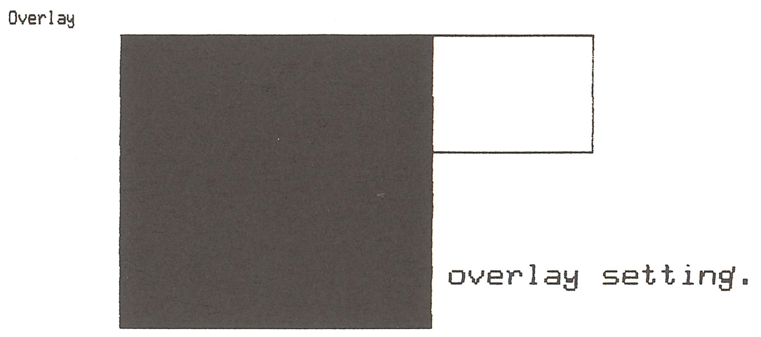
Overlay: modifies the display of an object that is positioned at the same location as another object. The overlaid object (the object defined with the overlay setting) is displayed according to the attribute settings that have been specified. The overlaid object affects the existing object only to the extent that it covers the existing object. The overlay setting is the same as laying a cutout on top of another cutout.

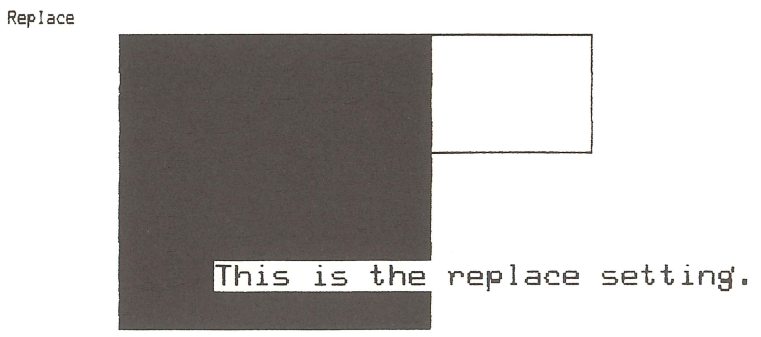
Attribute settings for the overlaid object will also modify the underlying object within the area of the overlaid object.

Text written with the overlay setting displays the text cell using the attribute settings of the background object.

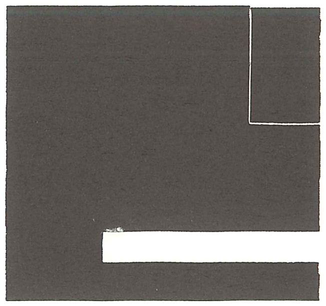
Examples

The examples below show a text string and a non-shaded box placed on top of and to the right of a shaded box. The text string and non-shaded box are written in the specified mode.

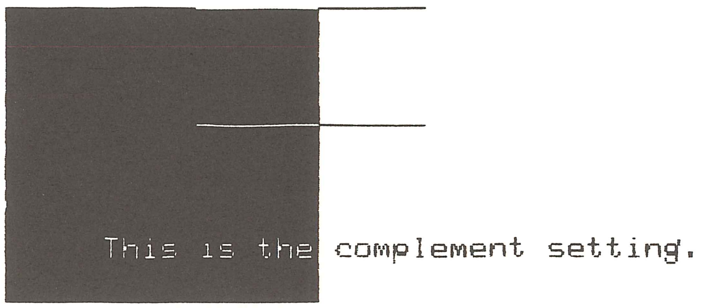




Erase



Complement



8 COMMANDS - REFERENCE

This chapter provides reference information about the typed commands used by the Graphics Editor. The typed commands are commands that you type on the main keypad. It will be helpful if you have already read Section 2, the Usage Section. Section 2 explains the use of the commands, the steps necessary to draw and edit pictures, and the use of files.

This chapter presents the commands in alphabetical order. The section for each command contains three parts:

- description
- format
- examples

The description contains the following information: a functional description of the command, the use of the command, and its use with other Graphics Editor commands and keys.

The format shows the syntax: the command, its arguments, and any optional arguments to the command. Optional arguments are shown in parentheses.

Examples show the use of the command and changes in screen display from command use.

CONVENTIONS

The conventions used in this chapter are:

Convention	Description
UPPERCASE	command
lowercase	required argument
( )	delineates an optional argument
●	current object
n	the relative number of an object in a picture (for example, 1 is the first object in the file)
,	separator to designate multiple objects
$	the last object in a picture
@	last address range used
*	all objects in a file (1,$)
label	identifying label applied to one or more objects
name	temporary label for an object
range	any of the above objects or addresses
RETURN	RETURN key on the main keypad

The Graphics Editor commands are:

- ALTER
- COPY
- DELETE
- EXTRACT
- FONT
- GROUP
- JOIN
- LABEL
- MOVE
- NAME
- QUIT
- READ
- SCALE
- TILT
- UPDATE
- VERIFY
- WRITE

ALTER

Function

The ALTER command sets an environment in which you can modify existing pictures. Using the ALTER command performs the same function as the ALTER key. However, the ALTER command allows you to specify the range to be modified. The default is the entire file.

After typing the ALTER command, the Graphics Editor displays a prompt, depending on the range you specified.

If you specified one object in the range, the graphics cursor is positioned on that object. You can then change any of the attribute settings (after pressing the ATTRIBUTE key) or use the interaction keys.

If you used the default or if you specified a range, the Graphics Editor displays a prompt to select the next, prev, or • key. The next key positions the graphics cursor on the next object in the file, the prev key positions the graphics cursor on the previous object in the file, and the • key returns you to command level without saving any changes.

After the graphics cursor is positioned on the object to be modified, press the ENTER key. The Graphics Editor then displays a prompt indicating that the current object can be modified. At this point, you can use any of the attribute keys (after pressing the ATTRIBUTE key) or the interaction keys to modify the picture. For more information about the use of the interaction keys (move, add, open/close, and delete keys) and the ATTRIBUTE key, refer to Chapter 7.

To save your changes in the copy of the file being used by the Graphics Editor, press the ENTER key. To save the changes on disk, use the WRITE command.

To return to command level without saving the changes, press the • key.

After modifying the picture, it may be necessary to use the UPDATE command to refresh the screen.

Format

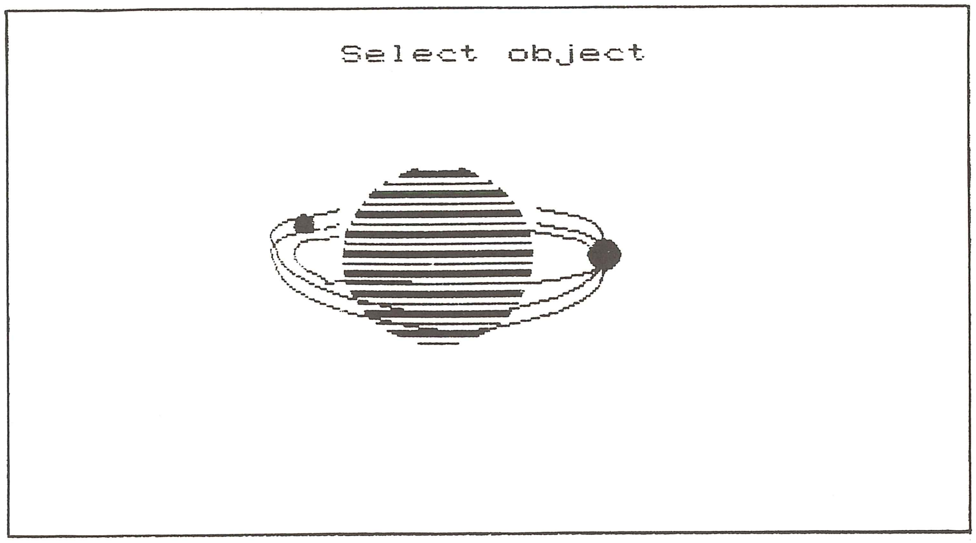
ALTER *(range)* RETURN

where range can be:

•	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	name assigned with GROUP, LABEL, or NAME commands

Example ALTER * RETURN

The example below shows the screen display after typing above command.



```
Select object
```

COPY

Function

The COPY command draws a copy of the specified range of objects. The copy is positioned at the same location as the current object or objects. In a sense, the new copy is placed on top of the currently existing range. Multiple copies of an object or range can exist simultaneously at the same screen position.

After you type the COPY command, the Graphics Editor displays a prompt requesting confirmation of the duplication. Press the ENTER key to draw a copy of the specified range. To return to command level without creating a copy, press any other key on the keyboard.

You can specify a range of objects to be copied. The default range is the entire file.

To place a copy of the object elsewhere, use the MOVE command. The MOVE command moves the original object to the new location and leaves the copy at the original location. The relative order of the objects within the file is changed to correspond to this.

To save any changes made with the COPY command on disk, use the WRITE command.

Format

COPY *(range)* RETURN

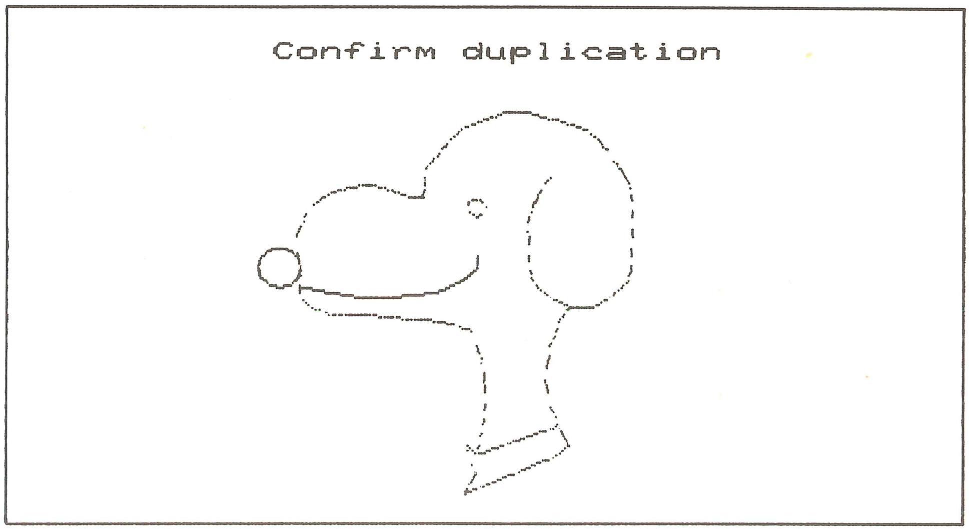
where range can be:

•	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	name assigned with GROUP, LABEL, or NAME commands

Example

COPY 1,4 RETURN

The example below shows the screen display after typing the above command and pressing the RETURN key. Press the ENTER key to confirm. The Graphics Editor displays the specified objects in dotted lines.



```
Confirm duplication
```

DELETE

Function

The DELETE command deletes all objects in the specified range. The current object is used as a default range if the range is not specified. After deleting an object, the Graphics Editor sets the current address to the object following the last deleted object.

To delete a screen erase in a file, specify a range in which the screen erase is included. The object before and the object after the screen erase must be included in the range for the screen erase to be deleted.

Format

DELETE *(range)* RETURN

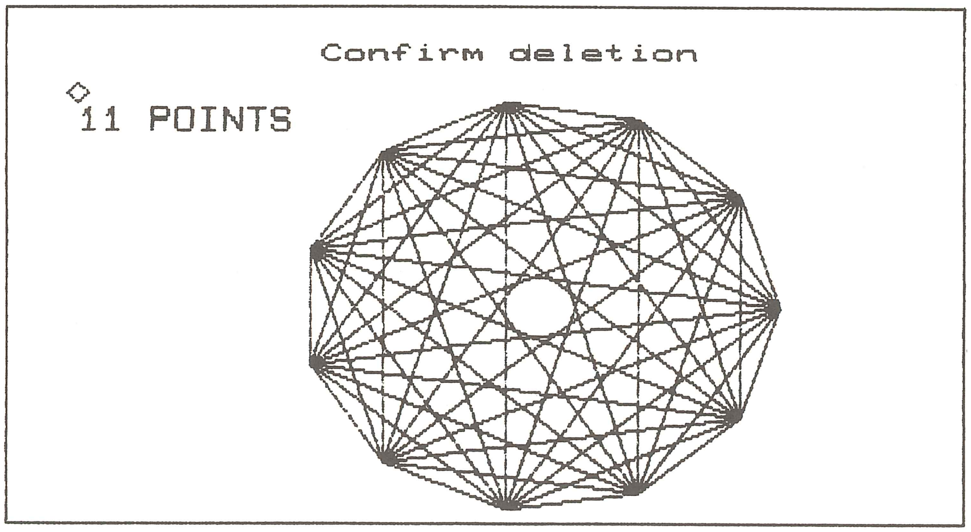
where range can be:

•	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	name assigned with GROUP, LABEL, or NAME commands

Example

DELETE 2 RETURN

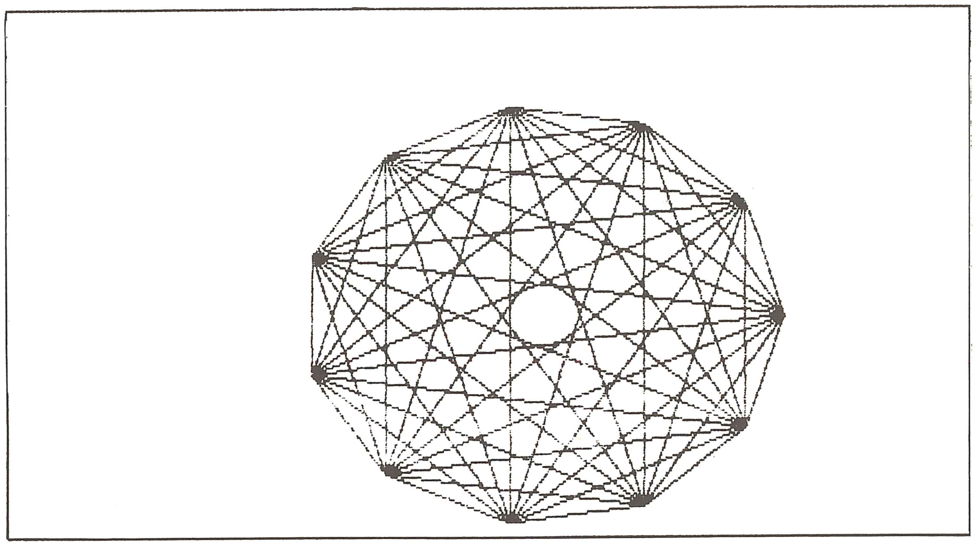
The example below shows the screen display generated by typing the above command and pressing the ENTER key. The graphics cursor is positioned on the specified object. Press the ENTER key to confirm the deletion.



```
Confirm deletion
◇
11 POINTS
```

continued next page

After confirming the change with the ENTER key, the display below is generated.



EXTRACT

Function

The EXTRACT command writes the specified range to disk in ASCII format. The range is written with y-coordinates being expressed as relative values, rather than absolute values. This provides a format compatible with other GIGI software.

Files containing the relative addresses are not read correctly in all cases by the Graphics Editor. WRITE should be used to write files to disk, if the files are to be used again with the Graphics Editor or the Slide Projection System. The EXTRACT command should not be used in such situations.

The default range is the entire file.

Format

EXTRACT *(range) filename* RETURN

where range can be:

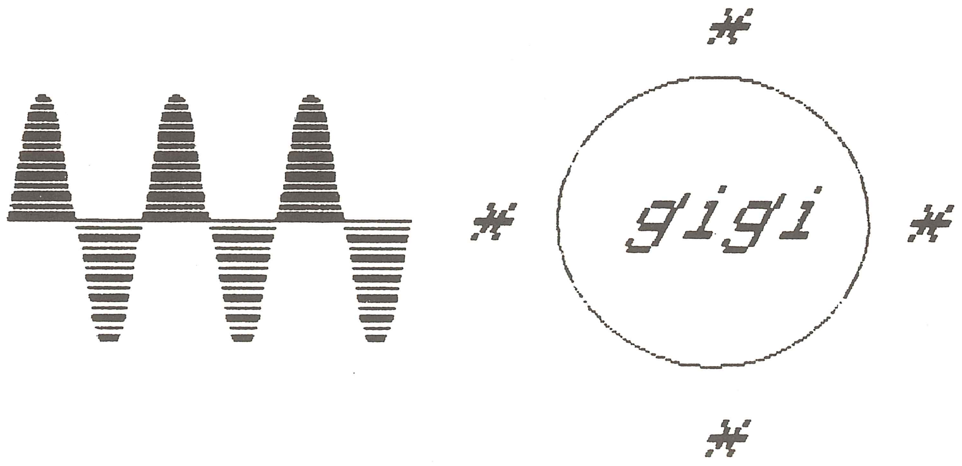
●	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	name assigned with GROUP, LABEL, or NAME commands

filename is the name of the file written to disk

Example

EXTRACT PIC.EXT * RETURN

Two file listings are displayed on the following pages. Both listings describe the same picture. The first file, PIC.EXT, was written with the EXTRACT command. The second file, PIC.LST, was written with the WRITE command. The picture described by these listings is shown below.



continued next page

PIC.EXT

```
w(v,n0,p1(m2),s0,m1,a0,i7)
t(a0,d0,s4,h6,i–27)
w(p3(m2),s1[,+26])s(i0)
p[34,+26]c(s)[]
  [59,–100]
  [84,+100]
  [109,+100]
  [134,–100]
  [159,–100]
  [184,+100]
  [209,+100]
  [234,–100]
  [259,–100]
  [284,+100]
  [309,+100]
  [334,–100][](e)
w(p1(m2))
p(b)
p[559,+0]c[636,–86]
p(e)
w(c,p1(m2),s0)
p(b)
p[515,–34]t'gigi'
p(e)
p(b)
p[565,–184]t'*'
p(e)
p(b)
p[715,–27]t'*'
p(e)
p(b)
p[565,+148]t'*'
p(e)
p(b)
p[390,–26]t'*'
p(e)
w(v)
```

PIC.PIC

```
w(v,n0,p1(m2),s0,m1,a0,i7)
t(a0,d0,s4,h6,i−27)
w(p3(m2),s1[,240])s(i0)
p[34,240]c(s)[]
  [59,140]
  [84,240]
  [109,340]
  [134,240]
  [159,140]
  [184,240]
  [209,340]
  [234,240]
  [259,140]
  [284,240]
  [309,340]
  [334,240][](e)
w(p1(m2))
p[559,240]c[636,154]
w(c,p1(m2),s0)
p[515,206]t'gigi'
p[565,56]t'*'
p[715,213]t'*'
p[565,388]t'*'
p[390,214]t'*'
w(v)
```

FONT

Function

The FONT command loads a character font into an alternate character set.

The character font must exist in a file on disk. Character fonts are created with the Character Set Editor or with ReGIS and a text editor. To use the character font after loading it into the alternate character sets, use the Font key to select the font.

Unused characters appear as shaded or blank text cells.

A character font remains loaded in the alternate character set until one of the following occurs:

- Another character font is loaded into the alternate character set.
- GIGI is turned off.
- The GIGI features are reset, according to methods described in the GIGI/ReGIS Handbook.

Format

FONT *font-number filename*

where:

font-number must be 1, 2, or 3. Font-number indicates the alternate character set into which the font-file is read.

filename is the name of the file that contains the ReGIS commands for the character descriptions.

Example

FONT 1 GREEK RETURN

GROUP

Function

The GROUP command assigns a label to the specified range of objects. The default range is the current object. A label can include one to ten alphanumeric characters, of which the first character must be alphabetic.

This label is written into the file when the WRITE command is used. This label is not visible when a picture is displayed. A GROUP label can be referenced in a range. Multiple GROUP labels can be used within a file.

These labels can be used for addressing by the Graphics Editor and are also recognized by the Slide Projection System.

The GROUP command identifies a range, while the LABEL command flags only a location. GROUP labels apply to all objects within the designated group. LABEL labels apply to all objects from the location of the label to the end of the file. NAME labels are temporary and are not written to the file with the WRITE command.

Format

GROUP *(range) label* RETURN

where range can be:

•	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	is a name including one to ten alphanumeric characters

Example

GROUP 2,4 TREE RETURN

JOIN

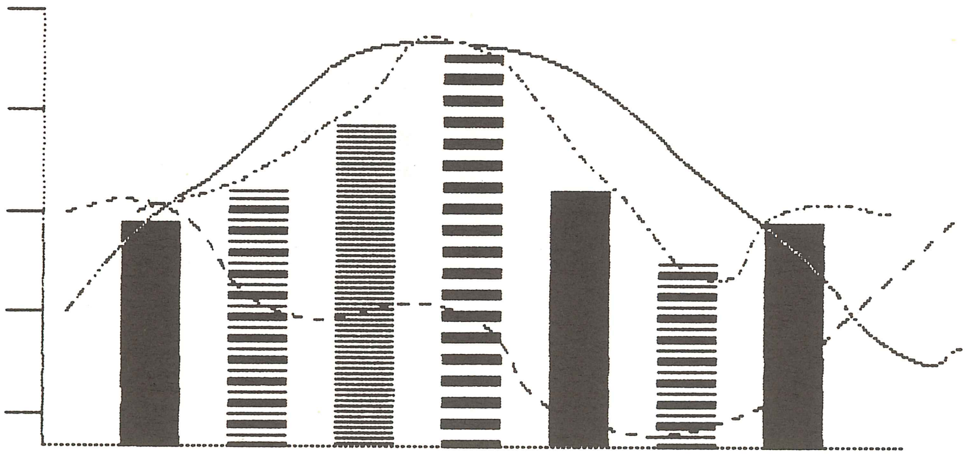
Function The JOIN command deletes the screen erases that form new pages. The first new page following the first address in the range specified is deleted.

Format JOIN *range*

where range can be:

●	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	name assigned with GROUP, LABEL, or NAME commands

Example JOIN 3,4 RETURN



LABEL

Function

The LABEL command flags the current object with the specified label name. This label is written into the file when the WRITE command is used. This label is not visible when a picture is displayed. A label can include one to ten alphanumeric characters, of which the first character must be alphabetic.

A LABEL label can be referenced in a range. Multiple LABEL labels can be used within a file.

The LABEL command flags only an object, while the GROUP command identifies a range. LABEL labels apply to all objects from the location of the label to the end of the file. GROUP labels apply to all objects within the designated group. NAME labels are temporary and are not written to the file with the WRITE command.

Format

LABEL *(range) label* RETURN

where range can be:

•	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	is a name including one to ten alphanumeric characters

Example

LABEL @ STAR RETURN

MOVE

Function

The MOVE command moves a range of objects to the specified location. This range is deleted from its original location when it is moved. The current location is set to the last object in the range that was moved.

You can specify the range to be moved. The default range is the current object.

After typing the MOVE command, the Graphics Editor prompts you for the current origin and the new origin.

The current origin indicates the fixed point or the central point for the range being moved. Position the graphics cursor to the central point by using the directional or the arrow keys. After positioning the graphics cursor, press the ENTER key.

The new origin is the location to which the range will be moved. Position the graphics cursor to the new location with the directional or the arrow keys. After positioning the graphics cursor, press the ENTER key. After the Graphics Editor moves the range, you are returned to command level.

To return to command level at any point while using the MOVE command, press any key other than the ENTER or the directional or arrow keys. No changes will be written to the file.

If you are moving a range created with the COPY command, the MOVE command moves the original object to the new location and leaves the copy at the original location. The relative order of the objects within the file is changed to correspond to this.

To refresh the screen after moving a range, use the UPDATE command. To save the changes on disk, use the WRITE command.

Format

MOVE *(range)* RETURN

where range can be:

.	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	name assigned with GROUP, LABEL, or NAME commands

SELECT ORIGIN

use the directional or arrow keys and press the ENTER key

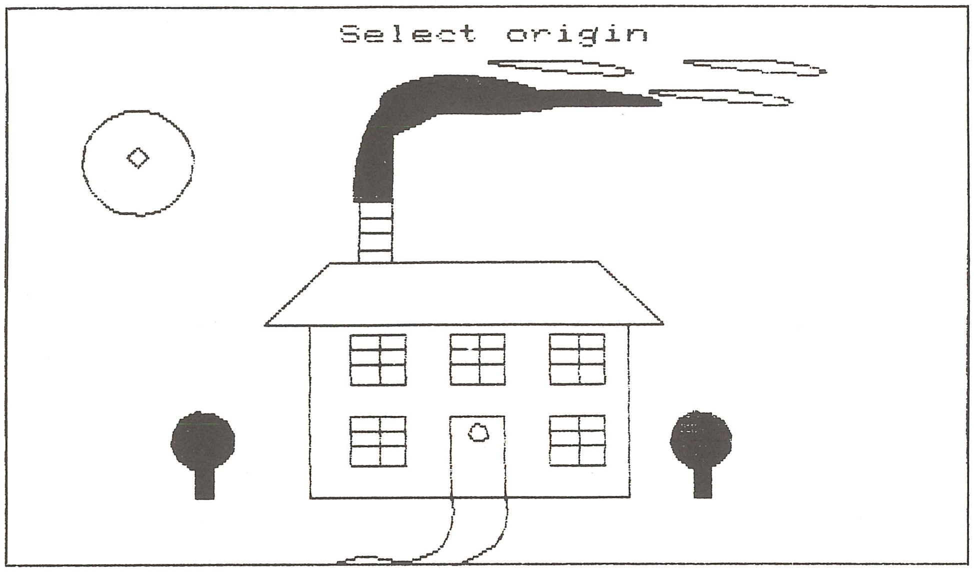
SELECT NEW ORIGIN

use the directional or arrow keys and press the ENTER key

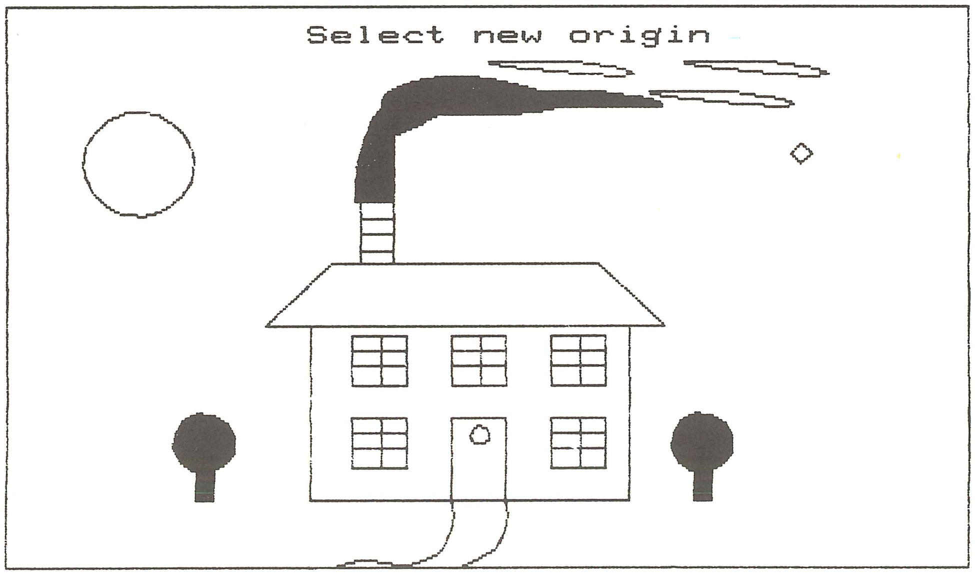
Example

MOVE 3 RETURN

In this example the screens show the displays generated with the MOVE command. The screen below shows the first display after typing the command shown above and pressing the RETURN key.

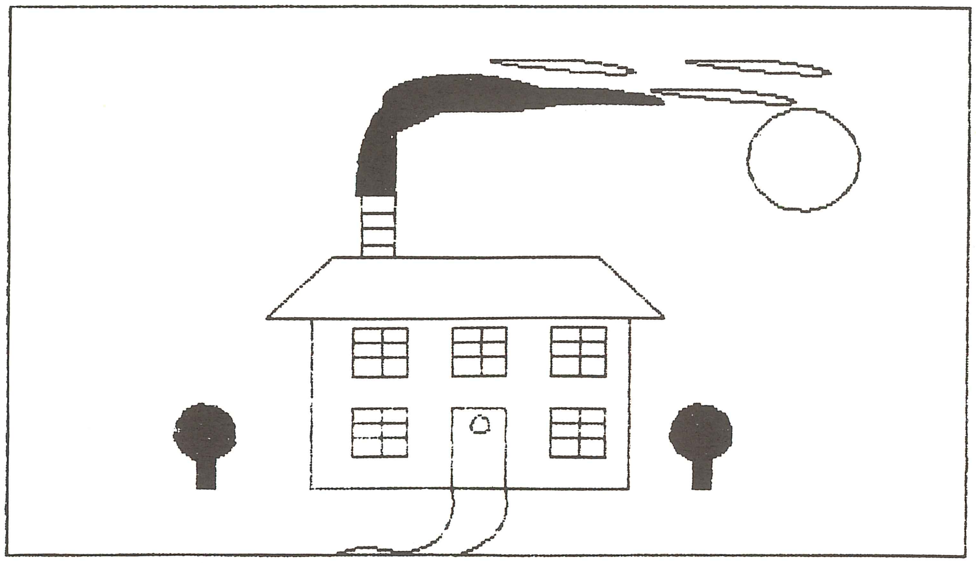


The screen below shows the display after repositioning the graphics cursor (with the directional keys) to the new location and confirming the change by pressing the ENTER key.



continued next page

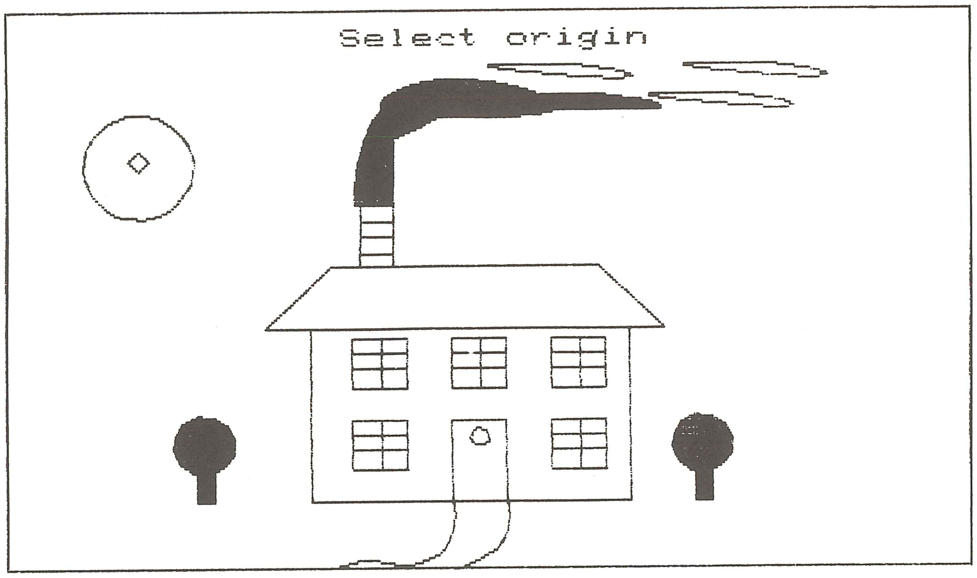
The screen below shows the display after confirming the change by pressing the ENTER key.



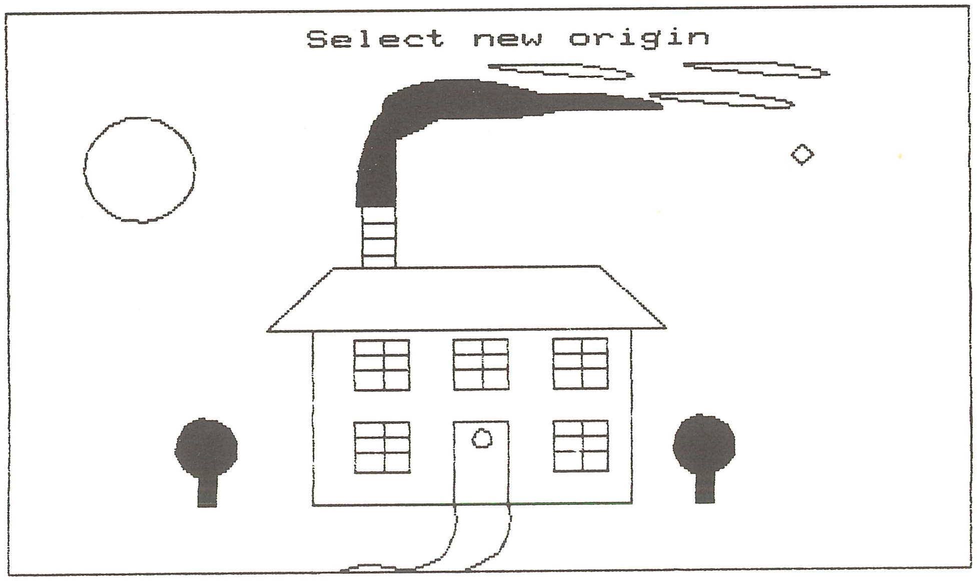
Example

MOVE 3 RETURN

In this example the screens show the displays generated with the MOVE command. The screen below shows the first display after typing the command shown above and pressing the RETURN key.

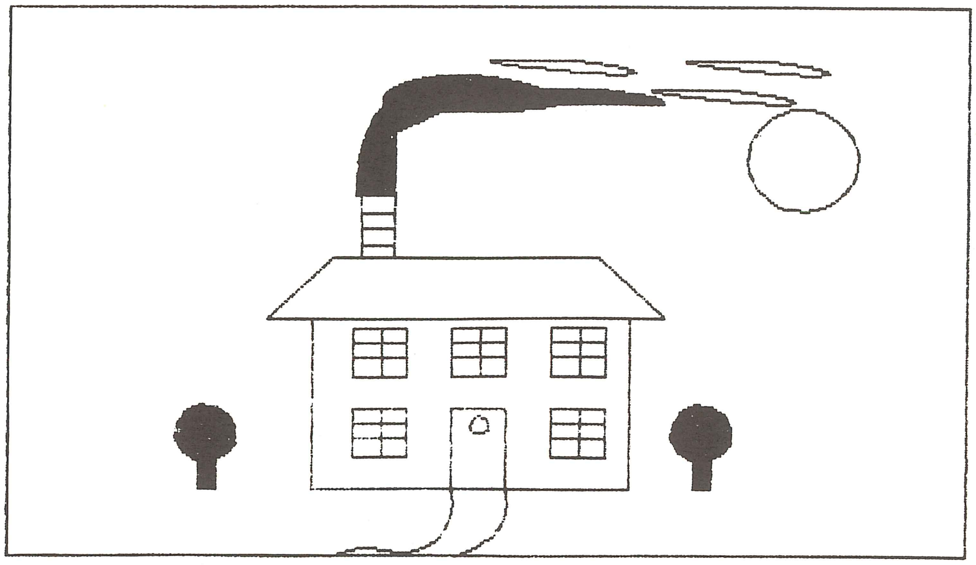


The screen below shows the display after repositioning the graphics cursor (with the directional keys) to the new location and confirming the change by pressing the ENTER key.



continued next page

The screen below shows the display after confirming the change by pressing the ENTER key.



NAME

Function

The NAME command inserts a temporary label at the specified location. This temporary label, which is useful for editing, flags a location in the GIGI file copy used by the Graphics Editor. The label does not display on the screen. The WRITE command does not write the label into the file on disk. A NAME label can be referenced in a range. Multiple NAME labels can be used within a file. A label can include one to ten alphanumeric characters, of which the first character must be alphabetic.

The NAME command is temporary and flags a location. The LABEL command also flags a location, but the label can be written to the file. LABEL labels apply to all objects from the location of the label to the end of the file. The GROUP command identifies a range and can be written to be file. GROUP labels apply to all objects within the designated group.

Format

NAME *(range) label* RETURN

where range can be:

•	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	is a name including one to ten alphanumeric characters

Example

NAME 8 BOX RETURN

QUIT

Function

The QUIT command stops the execution of the Graphics Editor and returns you to system command level. To save any changes, specify the WRITE command before using the QUIT command. The Graphics Editor does not save any changes or pictures when the QUIT command is used.

The QUIT command resets all SETUP parameters and ReGIS values to the default values.

Format

QUIT RETURN

Example

QUIT RETURN

READ

Function

The READ command reads the named file into the buffer used by the Graphics Editor. The file can then be displayed or edited. To be displayed and edited, the file must be a ReGIS file created with the Graphics Editor or a text editor.

Use the UPDATE command to display the file after it has been read in.

Multiple files can be in the buffer at one time. Unless otherwise specified, the Graphics Editor places the file being read in at the end of the last file. You can, however, position the files in a different order by using the graphics cursor to specify another sequence. To do this, position the graphics cursor by first using the ALTER key (to enter ALTER mode) followed by the next or prev keys to position the graphics cursor on a particular object. This object indicates where the next file will be read in. The order in which the files are read in determines the sequence in which the objects are drawn. The sequence is:

1. The objects (in the file already in the buffer) before the current position are drawn.
2. The objects in the file being read in are drawn.
3. The objects (in the file already in the buffer) after the current position are drawn.

Using the graphics cursor to select the current position affects only the sequence in which objects are drawn; it does not affect the screen positioning of any of the objects. The sequence in which objects are read in will affect the screen display if multiple objects are drawn at the same location.

Format

READ (•) *filename* RETURN

where (•) is the current position

filename is the name of the ReGIS file to be read in

Example

READ PICTUR.PIC RETURN

SCALE

Function

The SCALE command interactively modifies the size of objects.

When you type the SCALE command, the Graphics Editor displays prompts for you to select the origin, the reference point before the scale, and the reference point after the scale. You use the directional or arrow keys to move the graphics cursor to select these reference points.

The origin indicates the fixed point or the central point for the object being scaled.

The reference point before the scale represents the amount in the x and y directions that the object will be scaled.

The reference point after the scale represents the new location for the object after it is scaled.

You can specify the range. The default range is the current object, if no range is specified.

Format

SCALE *(range)* RETURN

where range can be:

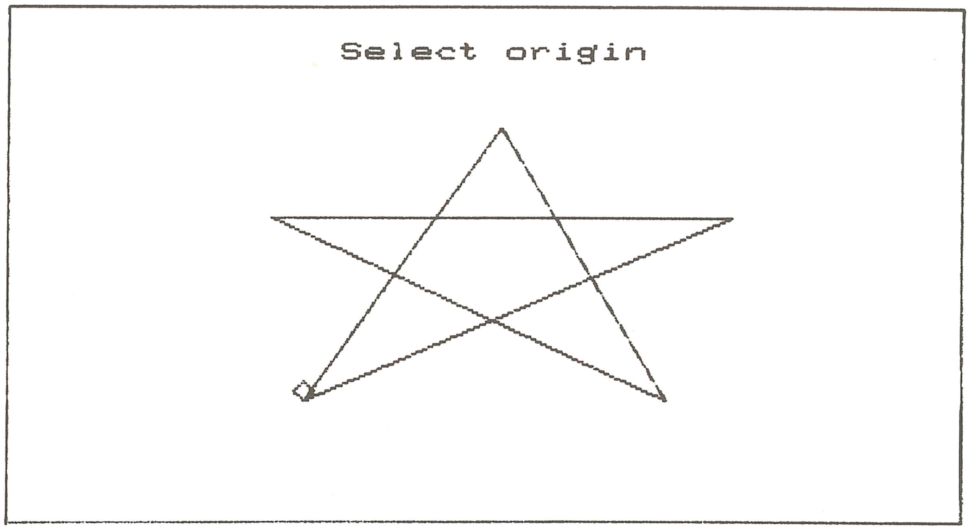
•	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	name assigned with GROUP, LABEL, or NAME commands

SELECT ORIGIN
use arrow or directional keys to move graphics cursor, end by pressing the ENTER key

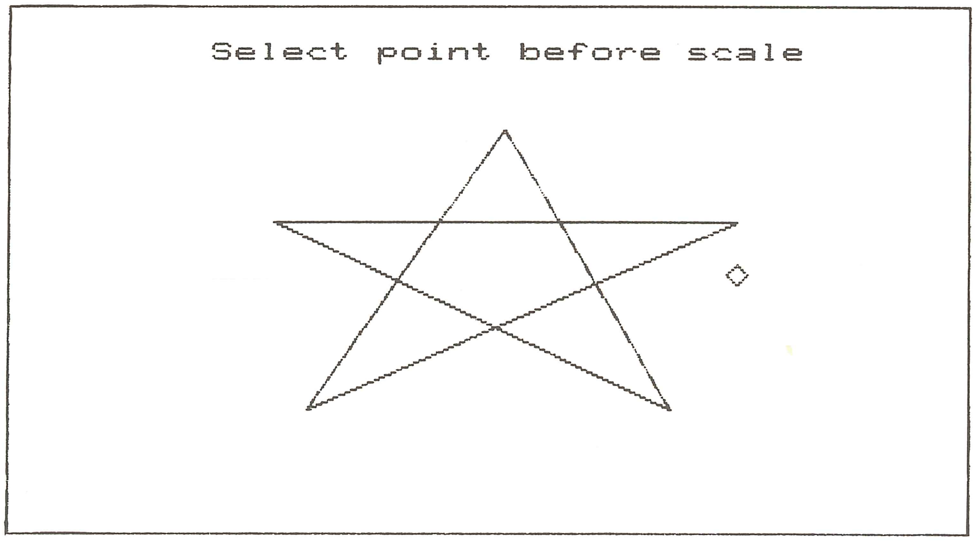
SELECT POINT BEFORE SCALE
use arrow or directional keys to move graphics cursor, end by pressing the ENTER key

SELECT POINT AFTER SCALE
use arrow or directional keys to move graphics cursor, end by pressing ENTER key

Example SCALE ● RETURN

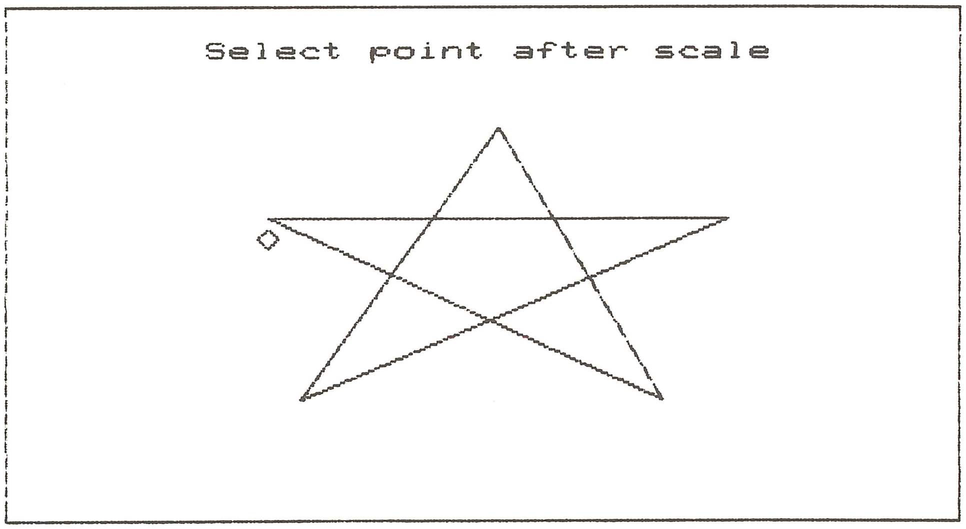


The screen below is generated after using the directional keys to reposition the graphics cursor to the origin of the scale and confirming by pressing the ENTER key.

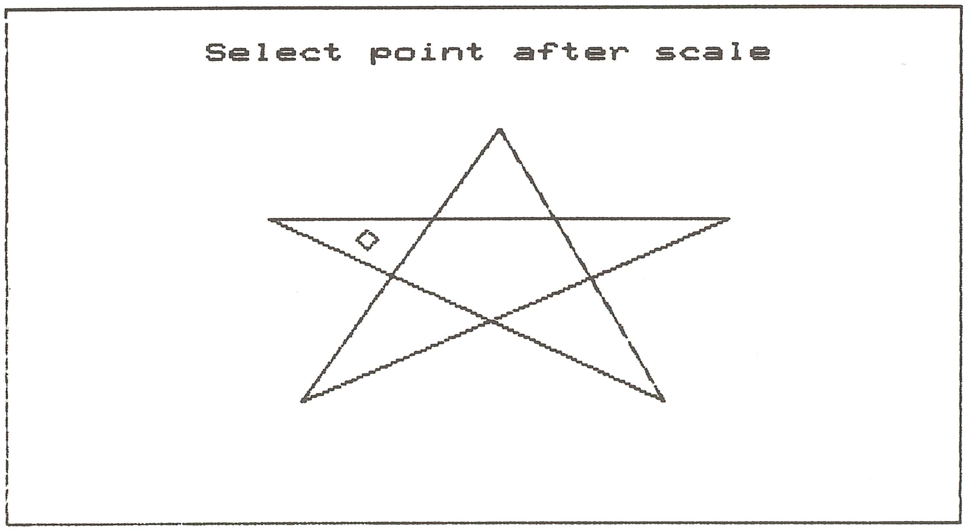


continued next page

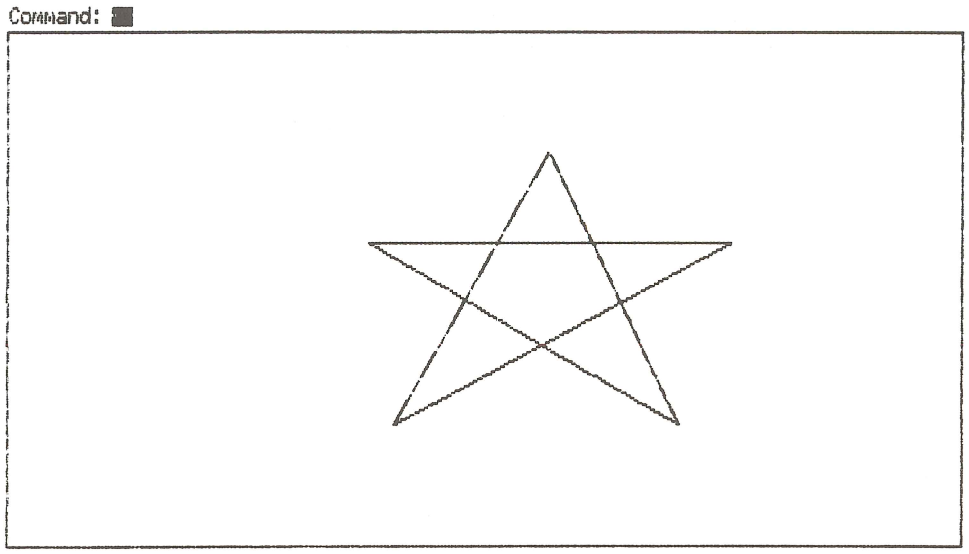
The screen below is generated after using the directional keys to select the point before the scale and confirming by pressing the ENTER key.


Select point after scale

The screen below is generated after selecting the point by moving the graphics cursor with the directional keys after the scale.


Select point after scale

The screen below is generated after confirming the final point by pressing the ENTER key and then typing the UPDATE command to refresh the screen.


```
Command:
```

TILT

Function

The TILT command rotates objects. After typing the TILT command, the Graphics Editor displays prompts for you to select the rotation center, the reference point before the rotation, and the reference point after the rotation. You use the arrow and directional keys to move the graphics cursor to select these reference points.

The rotation center indicates the fixed point or the central point for the object being rotated.

The reference point before the rotation represents the amount in the x and y directions that the object will be rotated.

The reference point after the rotation represents the new location for the object after it is rotated.

The Graphics Editor uses the current object as the default object, unless you specify one or more objects when you type the TILT command.

Format

TILT *(range)* RETURN

where range can be:

●	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	name assigned with GROUP, LABEL, or NAME commands

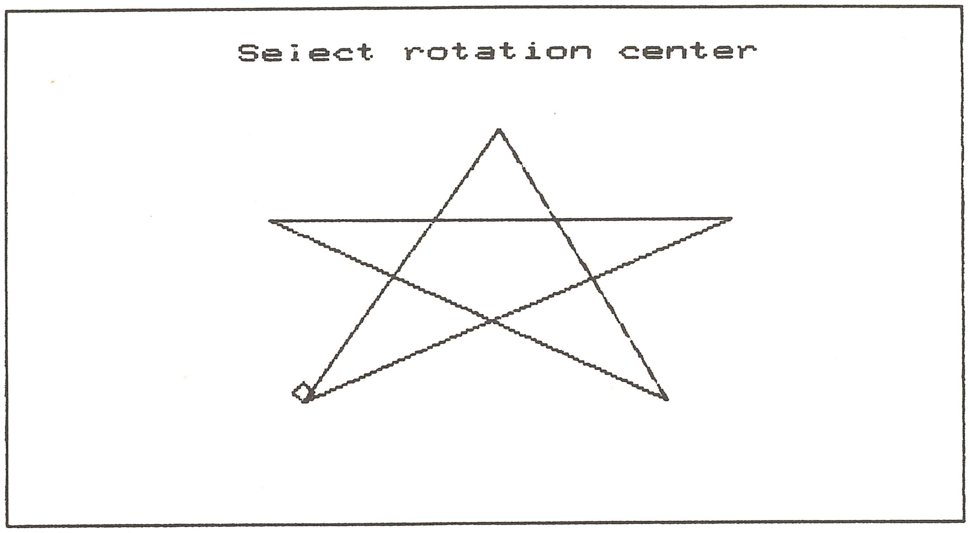
SELECT ROTATION CENTER
use arrow or directional keys to move graphics cursor, end by pressing the ENTER key

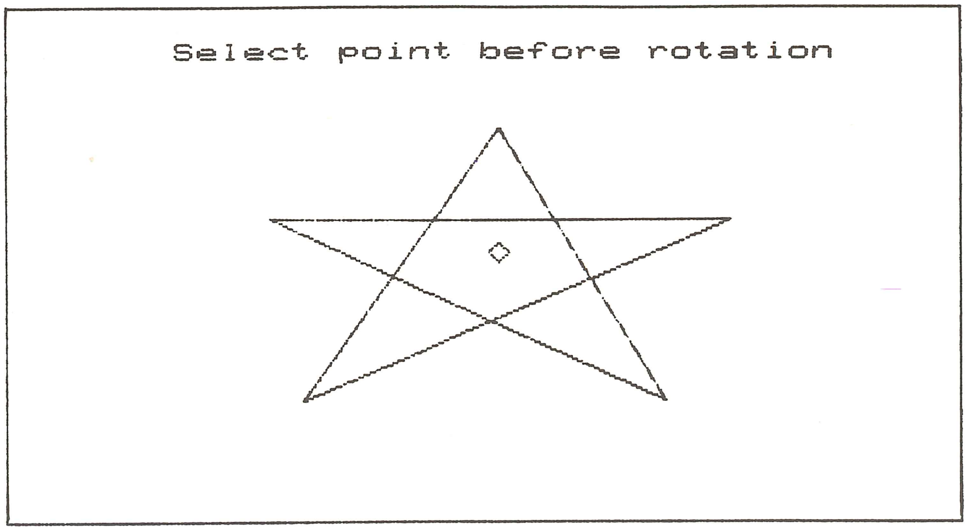
SELECT POINT BEFORE ROTATION
use arrow or directional keys to move graphics cursor, end by pressing the ENTER key

SELECT POINT AFTER ROTATION
use arrow or directional keys to move graphics cursor, end by pressing ENTER key

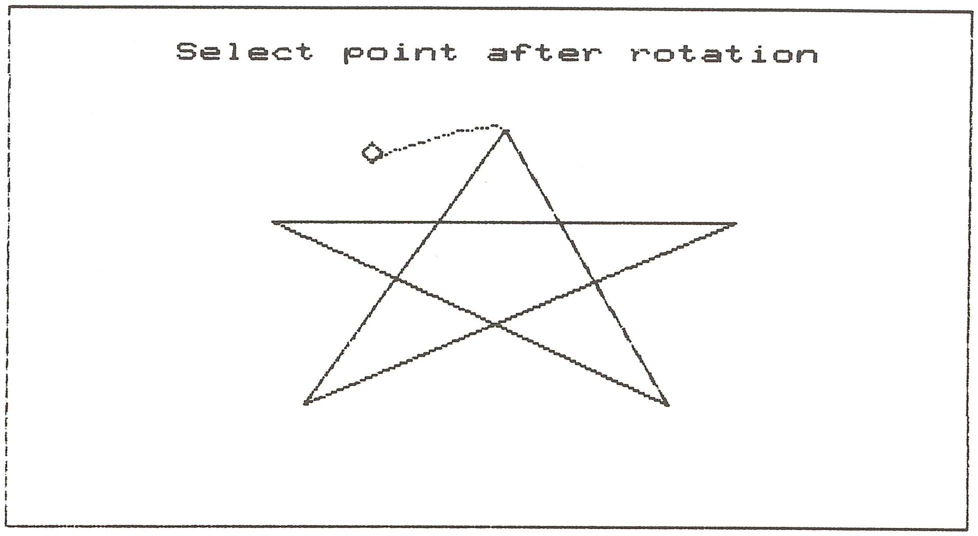
Example TILT 1 RETURN

The screens shown below are generated by typing the above command. The first screen is generated after typing the command and positioning the graphics cursor. The second screen is generated after confirming with the ENTER key and repositioning the graphics cursor. The third screen is generated after confirming with the graphics cursor and repositioning the graphics cursor. The fourth screen is generated after saving the change with the ENTER key and using the UPDATE command to refresh the screen.

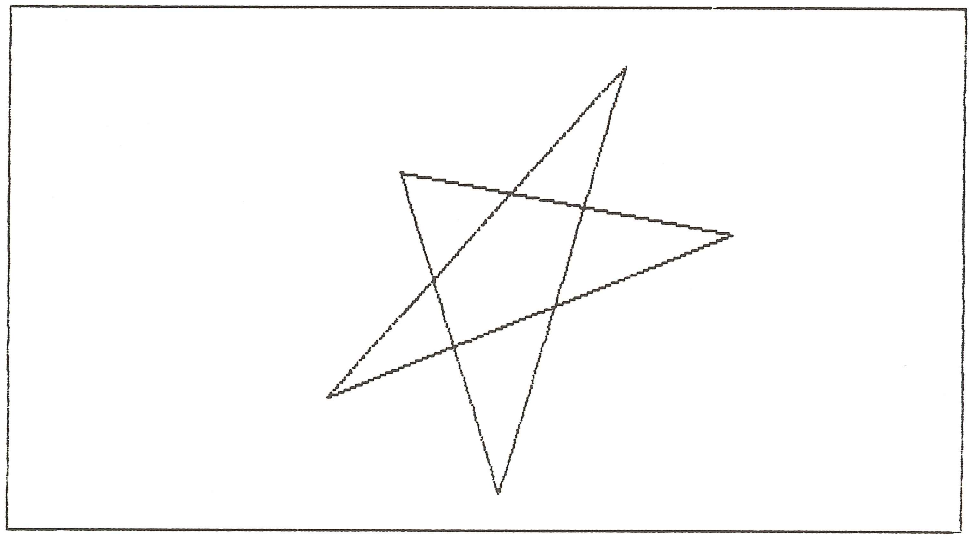

```
Select rotation center
```


```
Select point before rotation
```

continued next page


Select point after rotation

Command:



UPDATE

Function

The UPDATE command refreshes the screen display. Using the UPDATE command with no argument refreshes the entire screen. Specifying a range with the UPDATE command refreshes only the specified range.

The UPDATE command should be used to refresh the screen after reading files and modifying existing pictures.

Format

UPDATE *(range)* RETURN

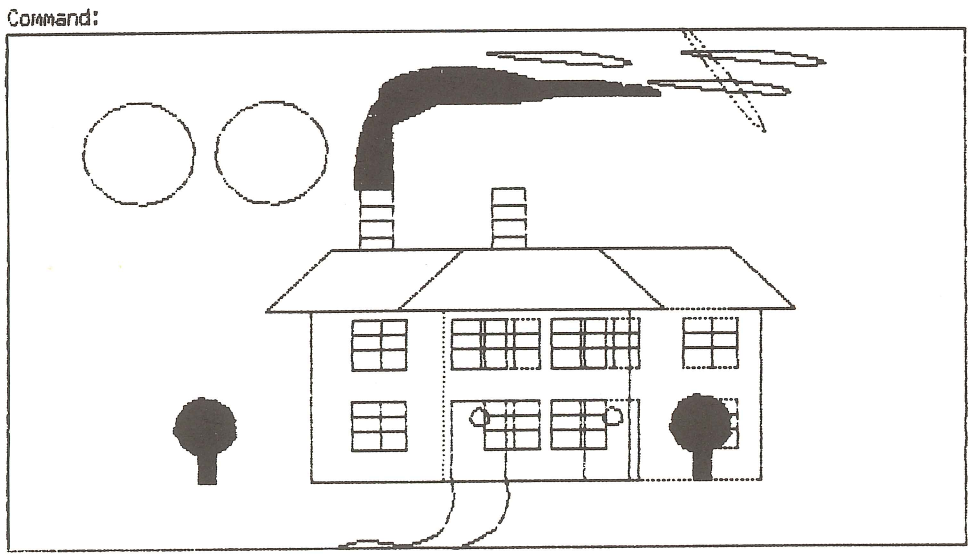
where range can be:

•	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	name assigned with GROUP, LABEL, or NAME commands

Example

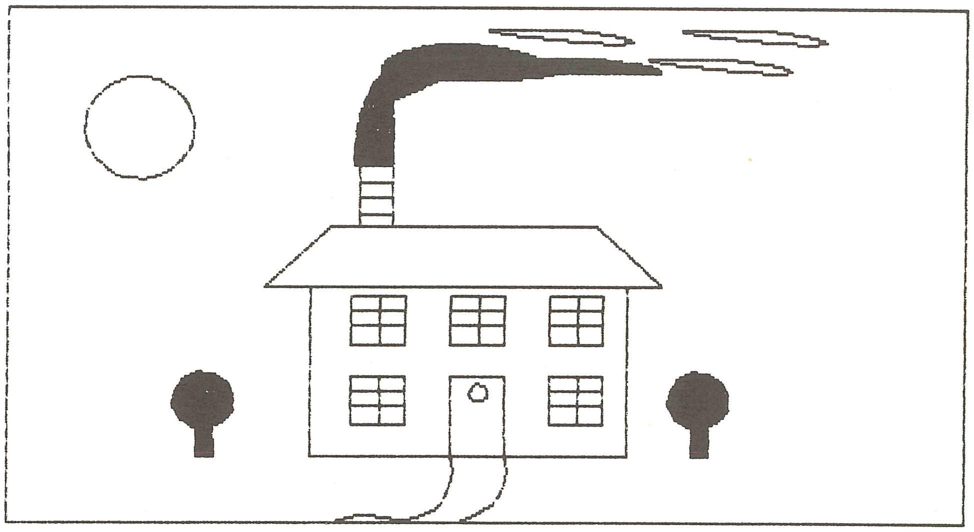
UPDATE RETURN

The examples below show a screen display before and after using the UPDATE command. The first screen display is generated after using the modifying a picture. The second screen shows a display after typing the UPDATE command.



continued next page

Command:



VERIFY

Function The VERIFY command displays a copy of the current file. The file is in the internal format used by the Graphics Editor. For more information, refer to Appendix A for a description of the format.

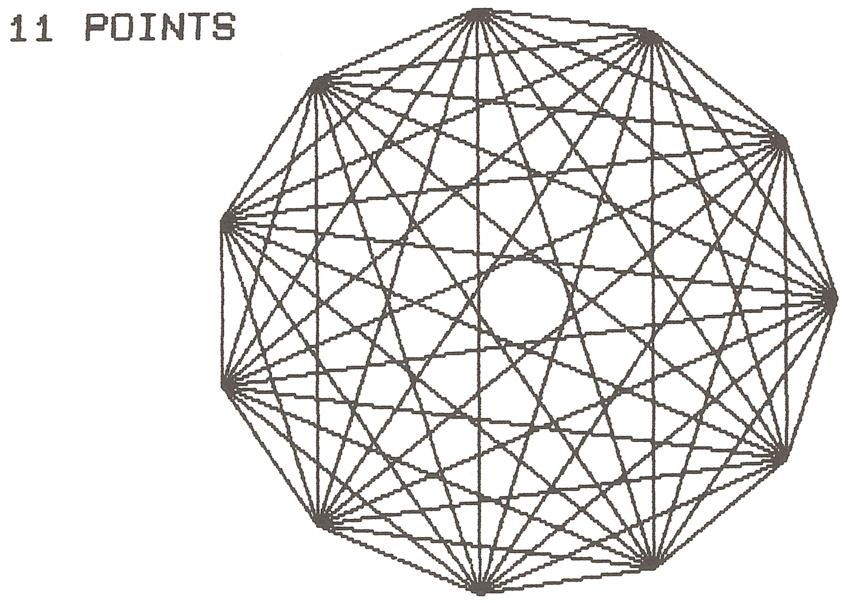
Format VERIFY *(range)* RETURN

where range can be:

•	current object
n	the relative number of an object in a picture
$	the last object in a picture
@	last address range used
*	the entire file (1,$)
label	name assigned with GROUP, LABEL, or NAME commands

Example VERIFY RETURN

The listing was generated by the display shown below.



continued next page

```
1:              Text Option (font) = 0
3:              Text Option (width) = 1
5:              Text Option (height) = 2
7:              Text Option (spacing) = 0
9:              Text Option (italic angle) = 0
11:             Text Option (slope angle) = 0
13:             Writing Option (raster op) = 3
15:             Writing Option (negate) = 0
17:             Writing Option (pattern) = 257
19:             Writing Option (pattern multiplier) = 2
21:             Writing Option (shading) = 0
23:             Writing Option (shading y axis) = 0
25:             Writing Option (pixel multiplier) = 1
27:             Writing Option (blink) = 0
29:             Writing Option (color) = 7
31:             Writing Option (background color) = 0
33:             Writing Option (raster op) = 0
35:             Writing Option (pattern) = 255
37:             Writing Option (color) = 4
39:             Writing Option (raster op) = 3
41:             Writing Option (color) = 7
43:             New Page.
44:             Writing Option (raster op) = 0
46:             Writing Option (color) = 4
48:             Text Option (width) = 2
50:             Text Option (height) = 3
52:      1      Text: [20,20] " 11 POINTS".
66:      2      Open Lines: [620,240].
180:            <End of File>
```

WRITE

Function

The WRITE command writes the ReGIS representation of the picture into a file on disk. All commands executed by the Graphics Editor, including screen erase, are written into the file on disk.

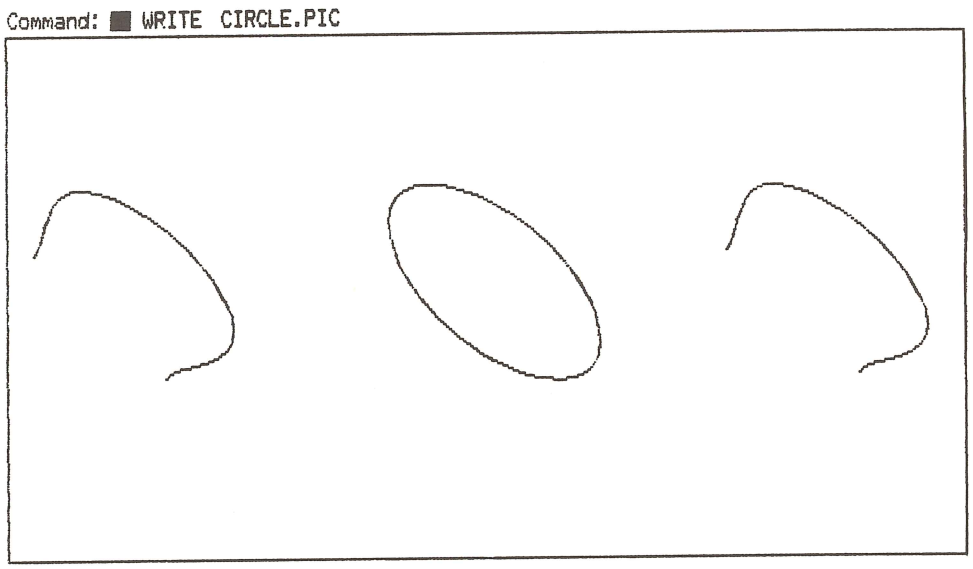
The file name can contain a maximum of six alphanumeric characters of which the first character must be a letter.

Format

WRITE *filename* RETURN

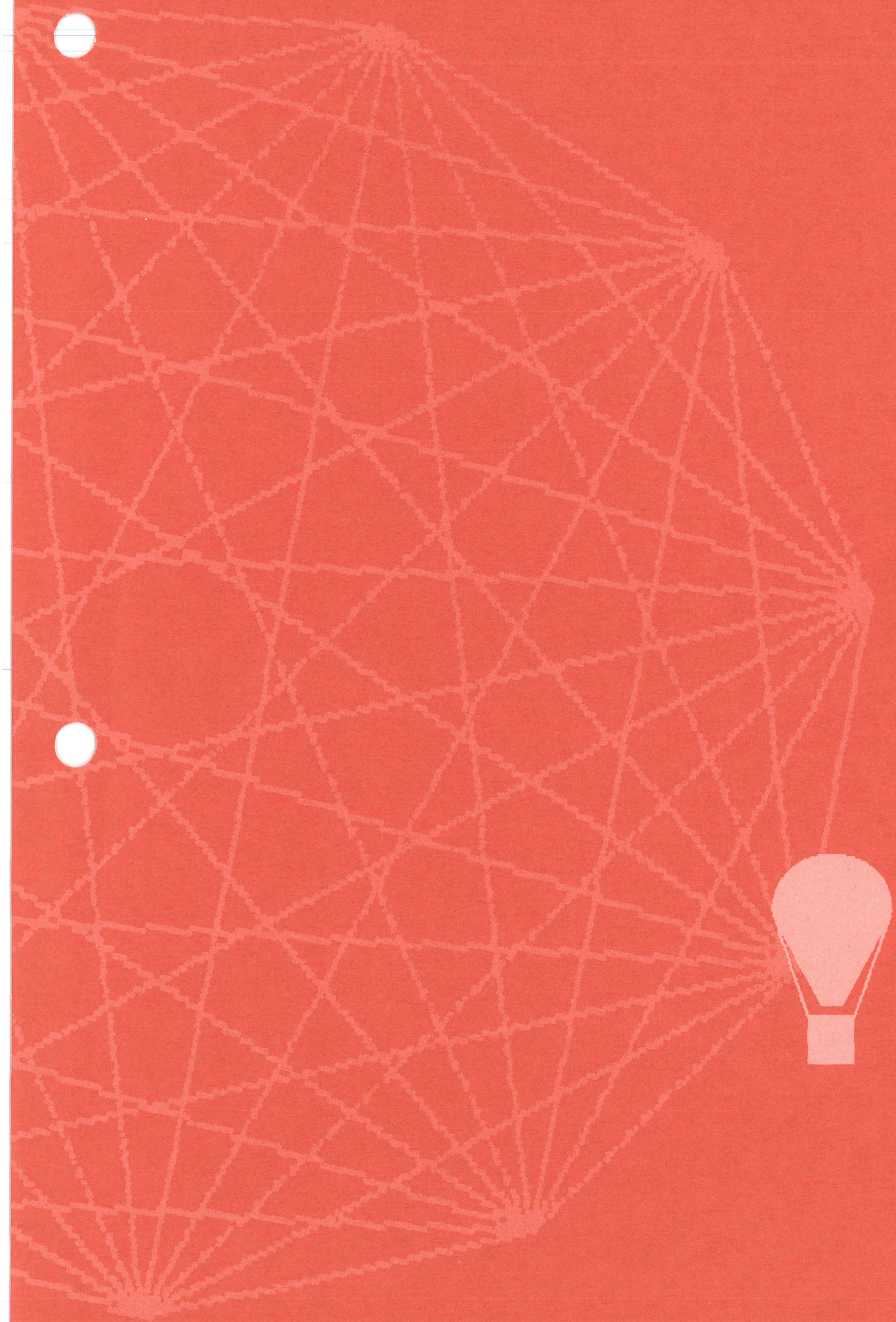
Example

WRITE PICTUR.PIC RETURN



```
Command: ■ WRITE  CIRCLE.PIC
```

Appendix and Glossary



APPENDIX: REGIS INPUT TRANSLATOR

GIGI Graphics Editor—ReGIS Input Translator

This is a technical description of the function of the ReGIS input translator section of the GIGI Graphics Editor package. It is assumed that the reader is already familiar with:

1. features and operation of the GIGI terminal
2. features and commands of the Graphics Editor
3. ReGIS language

The following notation conventions are used in this document:

1. Any literal text appears as such.
2. Examples of ReGIS commands appear in upper case.
3. Variable quantities are enclosed in angle brackets (e.g., <size>).

GLOSSARY

Label
A special form of ReGIS comment that lets the user of the Graphics Editor reference symbolically a position within a file of ReGIS commands. A label has the form:

;":<*labelname*>"

The <*labelname*> must be 1 to 10 alphanumeric characters, and the first character must be a letter.

Object
A subset of the ReGIS commands in a file. The concept of an object provides the user of the Graphics Editor with a convenient way of referencing a group of ReGIS commands. The format of an object in a ReGIS file is:

;":<*objectname*>{"<ReGIS commands>;"}"

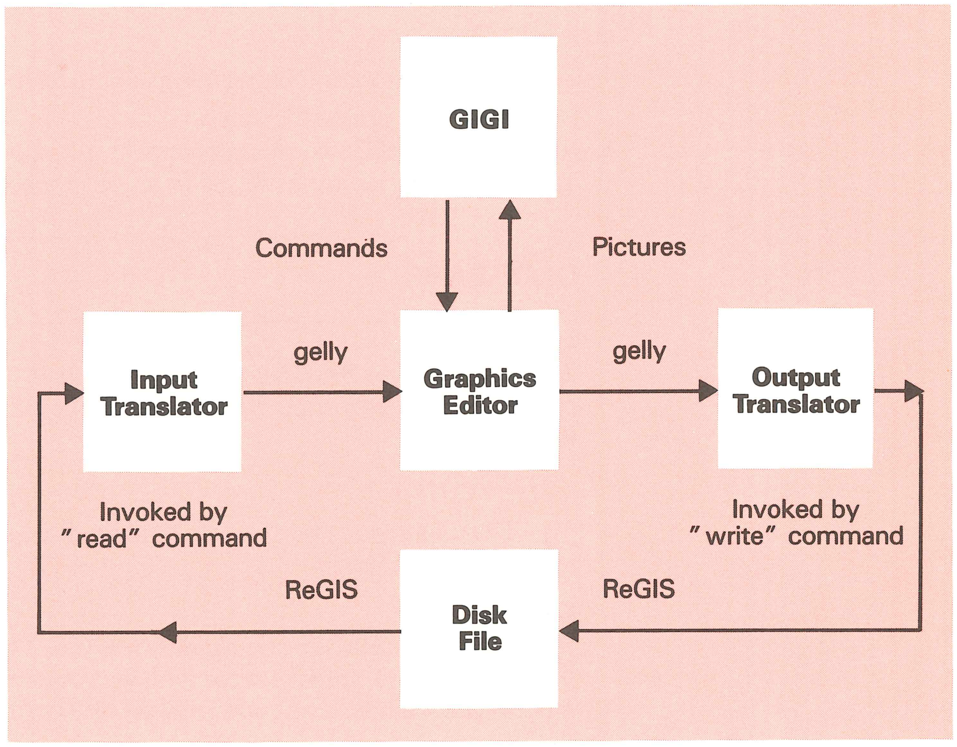
The <*objectname*> must be 1 to 10 alphanumeric characters, and the first character must be a letter. Objects may be nested, for example:

```
;":Object1{"P[100,100];":Object2{"C[+5];"}" V[150];"}"
```

Position Specifier
A ReGIS construction that specifies a cursor position. The cursor position can be either absolute or relative to the current cursor position. A position specifier is either a pixel vector (i.e., a digit from 0 to 7 inclusive) or a bracketed coordinate pair (e.g., [ ], [19, 73], [, +125]).

GRAPHICS EDITOR OVERVIEW

The GIGI Graphics Editor is a software package that lets a user create, store, retrieve, and update a graphic picture on the GIGI. For the purposes of this discussion, the reader can picture the Graphics Editor in the following fashion:



The user invokes the Graphics Editor to create a picture. Each picture is composed of one or more basic elements, known as "primitives". These primitives are:

1. Open lines
2. Closed lines
3. Open curves
4. Closed curves
5. Boxes
6. Circles
7. Text

In addition, the user may exercise control over various attributes, such as background and foreground colors, blinking, shading, character size, etc.

While creating or modifying a picture, the Graphics Editor stores picture information in an internal format known as "gelly" (pronounced "jelly"). When the user gives the "WRITE" command to save a picture on disk, the Graphics Editor translates its gelly to ReGIS, and stores it as ASCII text in a file on disk.

To retrieve a saved picture, the user gives the "READ" command, which instructs the Graphics Editor to convert the contents of the disk file from ReGIS to gelly, so that the user can manipulate the primitives that compose the picture.

The Graphics Editor does not restrict the user to reading pictures that were created by the Graphics Editor. The user can read any file that contains ReGIS commands, and the Graphics Editor will attempt to translate those commands to gelly. However, the nature of gelly is such that it is not as flexible as ReGIS; that is, there are some ReGIS constructions that cannot be represented in gelly.

This specification describes the ReGIS input translator (i.e., the section of the Graphics Editor that converts ReGIS to gelly). Hopefully, it will give the reader an insight into the capabilities of the Graphics Editor, and a knowledge of what elements of ReGIS can and cannot be manipulated by the Graphics Editor.

REGIS FILE FORMAT

Because the Graphics Editor is programmed in the FORTRAN language, it is assumed that the files that it reads have a record structure. The length of a record is limited to 130 characters. The Graphics Editor ignores any trailing space (blank) characters in a record.

GARBGELS

The term "Garbgel" applies to a ReGIS command that passes through the input translator unmodified. A Garbgel will be invisible to the user of the Graphics Editor, except for the fact that it is transmitted to the GIGI, and it will thus affect the picture as any ReGIS command would. Garbgels are created in either of two ways:

1. ReGIS commands that have no counterparts in the gelly language are converted to Garbgels. Examples of this are the entire L command and portions of the S command.
2. Any text that appears between the following pair of constructions

   ```
   ;".literal"
   ;"."
   ```

   is translated directly to a Garbgel (no other processing is done).

   When using the ;".literal" feature to generate a Garbgel, the user must take care not to disturb any of the current terminal attributes or parameters, such as writing options, cursor position, text options, screen options, etc. If this must be done, a non-destructive technique should be applied, for example:

   ```
   ;".literal" P(B)[200,200]BC(W(S1))[+50]P(E);"."
   ```

MACROGRAPH PROCESSING

When the Graphics Editor encounters a ReGIS macrograph definition (i.e., @:<*letter*>), it scans the following text until the end of the macrograph (@;) and stores the definition in its memory. When a reference is made subsequently to that macrograph (@<*letter*>), the Graphics Editor substitutes the current definition of the macrograph (if the macrograph is not defined, it is treated as if it contained no characters) for the @<*letter*> sequence.

When the Graphics Editor begins reading a ReGIS file, all macrographs are set to be undefined. Defining a macrograph more than once causes the old definition to be discarded and the new definition to take its place. The @. sequence sets all macrographs to be undefined.

It is not permitted to define or delete a macrograph from within a macrograph.

One macrograph may invoke another macrograph, for example,

```
@:AP[100,100]@B@;@BC[+50]@; @A
```

but recursion is illegal:

```
@:AP[100,100]@B@; @BC[+50]@A@; @A
```

There is a fixed amount of space available to the Graphics Editor for storing macrograph definitions. If this space is exhausted, the Graphics Editor gives a diagnostic message and continues processing; however, the results will probably not be what the user wanted. The space freed by redefining or deleting a macrograph is available for re-use.

C COMMAND (CIRCLES, CURVES)

The Graphics Editor translates the ReGIS C command to an Open Curves, Closed Curves, or Circle primitive. Open Curves and Closed Curves primitives are derived from (S)/(E) and (B)/(E) option pairs, respectively. Between every (B) . . . (E) and (S) . . . (E) pair there may be only position specifiers (i.e., [x,y]) and (A<*angle*>) and (C) options; anything else will terminate the current primitive and elicit the error message.

```
C(B) or C(S) terminated prematurely
```

When drawing a Closed Curve, the final cursor position is defined as that of the second point in the C command; thus a (B)/(E) pair with fewer than two intervening position specifiers will be rejected.

Since an Open Curve is determined by a minimum of four points (including the starting position), the Graphics Editor will reject a (S)/(E) pair with fewer than three intervening position specifiers. The final cursor position after drawing an Open Curve primitive is the position of the last point of the curve.

The output translator converts Open Curve primitives to the form,

```
P[x0,y0]C(S)[][x1,y1][x2,y2] . . . [xn,yn][](E)
```

Any C(S) . . . C(E) construction that does not contain the opening and closing null brackets (as shown above) will be distorted somewhat when it is translated to gelly.

A Closed Curves or Open Curves sequence begun in one C command may be continued in another C command, provided that, 1) there are no other commands between the two C commands, and 2) temporary writing options were not set in the first C command. An example of a continued C command is,

```
P[442,239]C(B)[442,200]C[502,119](E)
```

The Graphics Editor can accommodate up to 63 point specifiers between a (B)/(E) or (S)/(E) pair.

When drawing a circle with the initial position on the circumference (the C option) AND an arc length of other than 360 degrees, the Graphics Editor cannot determine the final cursor position. To ensure proper positioning, such circles should be followed by a P command to reorient the editor.

L COMMAND (LOAD CHARACTER SET)

Every L command is translated to a Garbgel. The Graphics Editor performs some syntax checking of the L command in the process.

P COMMAND (POSITION GRAPHICS CURSOR)

When the input translator encounters a P command, it updates its "idea" of where the cursor is. This will affect the position of any objects that are drawn subsequently.

If temporary writing options are set, the only one that has any effect is the pixel multiplier, P(W(M<number>)) . . .

R COMMAND (REPORT)

The R command is translated in its entirety to a Garbgel. It does not seem that this command would have much utility in a picture file.

S COMMAND (SCREEN OPTIONS)

When the screen is erased using the E option of the S command, the GIGI sets all foreground pixels to the current foreground color (as specified by the I option to the W command). If the S command contains both the W and E options, the temporary writing options become the permanent writing options (the GIGI does this too!).

The following pieces of the S command translate to Garbgels:

Screen jog	Sjjjjj
Set upper left corner	S[x,y]
Move screen	S[+-x,+-y]
The following options:	A, H, N, T

Use of the features of the S command mentioned in the list above may conflict with screen usage by the Graphics Editor. The user is cautioned in this respect.

T COMMAND (TYPE TEXT)

This is probably the command that will cause the most problems for users of the Graphics Editor. The Graphics Editor provides control of character width, character height, italic slant, writing direction, and spacing. There is a choice between two types of spacing: 8 (Mosaic) or 9 (Normal) columns per character. All characters have 10 rows. Users of the more sophisticated features of the T command may find that the Graphics Editor will distort the text. Beware!

To draw characters with Normal spacing, either of two methods may be used:

T(S<*size*>,H<*height*>)

The H<*height*> is optional. An S <*size*> option nullifies the effect of any previous H<*size*> options.

T(S[<*rsize*>,<*csize*>],M[<*cmul*>,<*rmul*>])

The following should be true:

<*rsize*> = <*cmul*> * 9

<*csize*> = <*rmul*> * 10

There is only one way to specify Mosaic spacing:

T(S[<*rsize*>,<*csize*>],M[<*cmul*>,<*rmul*>])

The following should be true.

<*rsize*> = <*cmul*> * 8

<*csize*> = <*rmul*> * 10

V COMMAND (DRAW VECTOR)

If a paired (B) and (E) are separated by only position specifiers (i.e., pixel vectors and bracketed coordinates), they will translate to a Closed Lines primitive. The rest of the cases become Open Lines. Appearance of any option (B,E,S,W) terminates the current primitive. A V command with no position specifiers is ignored.

A few examples:

$$\underbrace{\texttt{V(B)[+100,+100]}}_{\textit{Open}}\underbrace{\texttt{(B)[-442,+239][+127,+38](E)}}_{\textit{Closed}}\underbrace{\texttt{[60,20][25,22](E)}}_{\textit{Open}}$$

$$\underbrace{\texttt{V[507,+79][511,+79]}}_{\textit{Open}}\texttt{(W(I3))}\underbrace{\texttt{[531,+45][-27,106]}}_{\textit{Open}}$$

A Closed Lines or Open Lines sequence that draws a rectangle (it doesn't matter what order the sides are drawn in) is identified as a special case and translated to a Box primitive.

A Closed Lines or Open Lines sequence begun in one V command may be continued in another V command, provided that, 1) there are no other commands between the two V commands, and 2) temporary writing options were not set in the first V command. An example of a continued V command is,

```
P[98,477]V(B)[688,47]V[401,25](E)
```

The Graphics Editor can accommodate up to 63 point specifiers in a vector. The user can divide longer vectors into groups containing 63 or fewer points.

W COMMAND (SET WRITING OPTIONS)

All writing options are converted to their gelly counterparts. There are no special notes about the W command, or the W subcommand to the C, P, S, T, or V commands.

DIAGNOSTICS

When translating a file of ReGIS commands to gelly, the Graphics Editor can encounter various error conditions. Some of these cause diagnostic messages to be printed. The following is a list of the messages, along with their interpretations:

"Too much input — out of room"

The Graphics Editor has a finite amount of internal storage for picture information. This space has been exhausted. The user should attempt to make the picture file smaller if possible.

"Illegal [x,y] coordinate specification"

A syntactical error appeared in a bracketed coordinate specification.

"Macrograph defined or deleted within a macrograph"

Deleting or defining a macrograph from within a macrograph is proscribed.

"Illegal character after @"

The Graphics Editor was not processing a macrograph definition, and it encountered an at-sign, and the character following the at-sign was something other than a letter, a dot (.), or a colon.

"Attempt to define non-alpha macrograph"

The sequence of characters

@:⟨*char*⟩

appeared, and ⟨*char*⟩ was not a lower or upper case letter.

"Macrograph storage exhausted"

The internal macrograph storage area within the Graphics Editor has been exhausted. The user should reduce the number and/or length of the macrographs in the ReGIS file.

"Macrograph calls nested too deeply"

Macrograph calls were nested more than 26 deep. This could be the result of a circular definition, for example:

```
@:AP[100]@B@;   @:BV[200]@A@;   @A
```

"Illegal syntax in L command"
Illegal syntax in R command
Illegal syntax in S command
Illegal syntax in V command
Illegal syntax in P command
Illegal syntax in C command
Illegal syntax in T command
Illegal syntax in W command
Illegal syntax in P option of W command

All errors of the "Illegal syntax" breed indicate some sort of indecipherable ReGIS was encountered.

"Fewer than 2 points in closed curve"

It is an error to have a closed curve sequence with fewer than two points between the (B) and (E). For more information, refer to the description of the C command.

"Fewer than 3 points in open curve"

It is an error to have an open curve sequence with fewer than three points between the (S) and (E). For more information, refer to the description of the C command.

"C(B) or C(S) terminated prematurely"

While processing an Open Curve or Closed Curve sequence in the ReGIS file, the Graphics Editor encountered one of:

(B) option in the C command

(S) option in the C command

(W) option in the C command

A new command other than C

The Graphics Editor issues the warning message and generates a Curve primitive that contains all the points that were seen so far. This in turn can evoke one of the diagnostic messages about having too few points in the curve.

"Illegal label or object name"

Object names and labels must be from 1 to 10 alphanumeric characters, and the first character must be alphabetic. The Graphics Editor encountered an object name or label that did not fit these requirements.

" ;" "}" " found and no object was open"

A ;"}" was encountered, and there was no prior unterminated ;"⟨*object*⟩{".

"Eof hit and open object(s) exist"

Not all objects begun with ;"⟨object⟩{" were closed with ;"}".

"Putbak error — not your fault"

This error indicates a program malfunction. Contact DEC.

"Too many points in line"

"Too many points in curve"

The Graphics Editor imposes a limit of 63 points in a curve or a line (not counting the initial position of the cursor before the V or C command). This limit was exceeded.

GIGI GLOSSARY

This glossary includes a list of terms used in reference to GIGI. For more information about particular subjects, refer to the manuals described in the preface.

alternate character set: one of the three user-definable character sets in GIGI; character sets can be loaded into the alternate character sets.

arrow keys: keys at the right of the main keypad marked with arrows and used, in general, to move the graphics cursor.

ASCII character set: the ASCII characters that are always loaded into GIGI. This is the default character set for GIGI.

attribute level: the level invoked by using the ATTRIBUTE key. Any of the attribute keys can be used to change attribute settings. The attribute keys are Bkgrd Color, Blink, Color, Font, Italic, Height, Negate, Pat Mult, Pattern, Shading, Slope, Spacing, Width, Write-mode.

attributes: a characteristic or distinctive feature; GIGI has screen, writing, and text attributes.

auto-repeat: the terminal feature that causes the continuous transmission of the character for as long as you press the key for that character.

auxiliary keypad: the section of the keyboard to the right of the main keypad consisting of the numeric keypad and the program function keys. The Graphics Editor and Character Set Editor each use an overlay to indicate the key functions when the Graphics Editor or the Character Set Editor is used.

auxiliary keypad overlay: a plastic cover for the auxiliary keypad which you use to relabel the keys.

BASIC interpreter: the firmware in GIGI that processes GIGI BASIC statements.
brightness: monitor control used to adjust the intensity of the screen.

character: an 8-bit ASCII code; displayable characters are represented by an eight-by-ten dot pattern that is the basic unit in any of GIGI's character sets.

character cell: an eight-by-ten cell which contains a character pattern.

character pattern: the dot pattern contained in a character cell.

character set: a group of characters (up to a maximum of 128) of which 95 are displayable.

character size: the attributes defining the width and height of a character.

color: an attribute that defines the screen or writing color.

command keyletter: a letter that directs the ReGIS interpreter to perform graphics operations.

command level: the level at which the cursor is blinking beside the command line. The function keys, the control keys, and the commands can be used at command level.

commands: words that can be typed when the Graphics Editor is at command level.

complement writing: the writing mode in which the writing color is complemented. The writing color is changed to the background color. The use of a pattern causes the background-colored part of the pattern to assume the writing color.

computer assisted instruction (CAI): use of a computer to augment the individual instruction process by providing the student with programmed sequences of instruction under computer control, permitting students to progress at their own rate.

control level: the level invoked after control keys are used. The control keys are ATTRIBUTE, ALTER, ENTER, and the • key.

current character set: the most-recently specified alternate character set.

current location: a location, maintained internally in GIGI, which is a pointer to the location last moved to or drawn to.

cursor: block cursor GIGI displays when not in graphics mode. See also graphics cursor.

default: a value assumed when no specific choice is given by the user or a program.

directional keys: keys on the auxiliary keypad that move the graphics cursor. These keys are printed on the Graphics Editor keypad overlays.

drawing: the operation GIGI performs to display lines or text on the screen.

erase writing: the writing mode in which previously drawn objects are erased.

font file: a file that contains the definition of a character or group of characters.

function level: the level invoked after using a function key. The attribute, control and interaction keys are active. The function keys are LINE key, BOX key, CIRCLE key, CURVE key, and ERASE key.

GIGI: Graphics Image Generator and Interpreter.

graphics cursor: a diamond-shaped cursor displayed when GIGI is in graphic mode. See also cursor.

graphic mode: terminal operating mode in which the ReGIS interpreter is enabled.

graphic text: the text displayed when GIGI is in graphic mode.

gray scale: 8 levels of intensity GIGI uses on black and white monitors.

host system: computer system (hardware and software) to which GIGI is connected. The software packages are used on host systems.

image: all objects displayed on the screen.

image memory: a bit map GIGI uses to maintain and display images on the screen.

interaction level: the level invoked by using the interaction keys. To use interaction keys, a function key must first have been used. The interaction keys are add, delete, move, next, open/close, prev, the arrow keys, and the directional keys.

italic: character slant.

keyboard: GIGI's main keypad and the auxiliary keypad.

keyboard overlay: a plastic cover for the keyboard that lets you relabel the keys.

label: an identifier used to distinguish an object or range of objects within a picture.

line pattern: an eight-dot pattern GIGI uses to write.

location: a point, defined with ReGIS statements, on the screen.

main keypad: the portion of the keyboard consisting of the alphanumeric characters.

mark: a displayed symbol used to specify a location.

monitor: a separate video device containing the cathode ray tube (CRT) GIGI uses to display screen images.

mosaic: a multi-character image created with characters from user-defined character sets. The spacing between characters is decreased.

multiplication factor: a number used to multiply the pixels in either a writing pattern or a text cell before displaying the pattern or cell.

negative writing: the reversed interpretation of the dot pattern GIGI uses for writing.

object: a distinguishable portion of the image, for example a single pixel which is a part of a line, or a line which is part of a drawing.

overlay writing: the mode of writing in which the overlaid object (the object defined with the overlay setting) is displayed over the existing object. The overlaid object affects the existing object appears to cover the existing object. The overlay setting is the same as laying a cutout on top of another cutout.

page: each image within a picture. A picture can contain one or many pages or images. A new page is created with the ERASE key.

picture: any combination of drawings and text displayed on GIGI's screen.

picture files: files that contain the ReGIS descriptions of pictures.

pixel: picture element; the smallest displayable unit on the monitor screen.

point: a single location on the monitor screen.

program function keys: the top four keys on the auxiliary keypad, labelled PF1, PF2, PF3, and PF4.

range: a single object or a group of consecutive objects with which the first and last objects are specified.

relative location: a point on the screen measured from the current location rather than the screen location.

ReGIS: Remote Graphics Instruction Set

replace writing: the mode of writing that causes the object or text string to be written on the screen and to replace any other object that has been written in the same location.

reset: to return to a known default condition.

reverse video: the reverse interpretation of the screen and image bits in video memory; screen color becomes drawing color and drawing color becomes screen color.

screen: the portion of the video monitor on which images are displayed.

scrolling: the continuous horizontal or vertical movement of objects to make room for new objects.

self-tests: internal tests of terminal hardware that GIGI performs.

set-up mode: GIGI operating mode in which set-up parameters can be changed.

set-up parameter: terminal characteristics that can be changed in set-up mode to adapt the terminal to the operating environment.

shading: an attribute that uses the writing color to shade or color in the object being drawn.

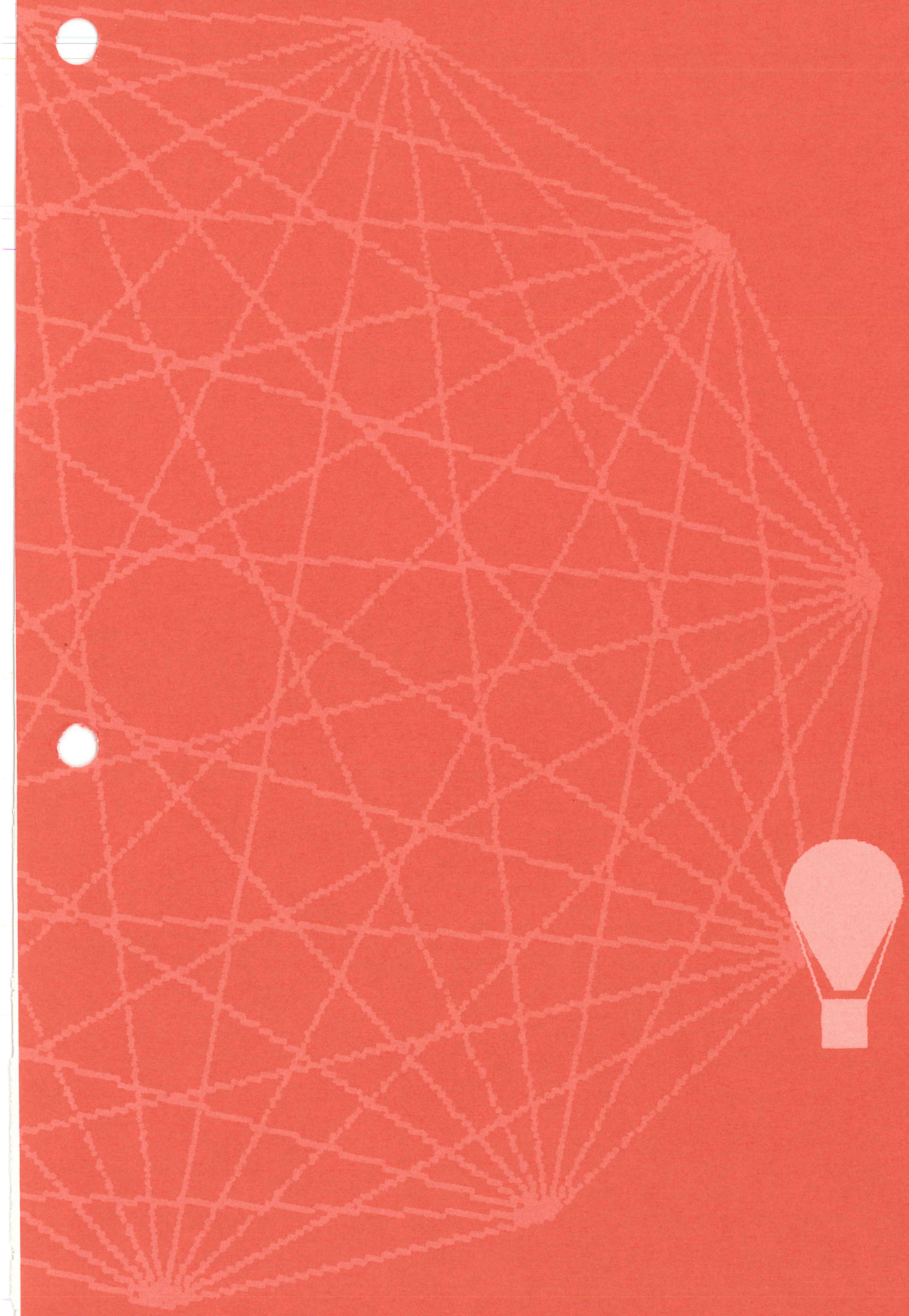
standard character set: the 128 ASCII character set of which 95 are displayable in graphics mode.

text: GIGI operating mode in which GIGI operates as a VT100- or VT52-compatible system terminal.

tray files: files that contain the names of one or more picture files.

writing: the operation GIGI performs to display lines or text on the screen.

Index



INDEX

GIGI Graphics Editor Manual
(AA-J942A-TK)

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an SPR form.

- Did you find errors in this manual? If so, specify by page.

- Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

- Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

- Please indicate the type of user/reader that you most nearly represent.

☐ Assembly language programmer
☐ Higher-level language programmer
☐ Occasional programmer (experienced)
☐ User with little programming experience
☐ Student programmer
☐ Non-programmer interested in computer concepts and capabilities

Name ______________________ Date ______________________

Organization ______________________

Street ______________________

City ______________________ State ____________ Zip Code or Country ____________

ase cut along this line.

Fold Here---

Do Not Tear — Fold Here and Staple--

No Postage
Necessary
If Mailed In The
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.152 MARLBOROUGH, MA 01752

Postage will be paid by:

digital

Education Computer Systems
200 Forest Street MR1-1/E47
Marlborough, Massachusetts 01752