AT&T

306-143
Issue 1

5620 Dot-Mapped Display

Release 2.0

Text/Graphics Guide

TRADEMARKS

The following trademarks are used in this manual:

- DOCUMENTER'S WORKBENCH — Trademark of AT&T Technologies, Inc.
- UNIX — Trademark of AT&T Bell Laboratories
- TELETYPE — Registered trademark of AT&T Teletype Corporation

CONTENTS

Chapter 1. INTRODUCTION

Chapter 2. CIP GRAPHICS PROGRAM

Chapter 3. PROOF PHOTOTYPESETTER SIMULATOR

Chapter 1

INTRODUCTION

Chapter 1

INTRODUCTION

THIS GUIDE AND YOU

This guide explains operation of the TELETYPE* 5620 Dot-Mapped Display (DMD) terminal when using the Text/Graphics Package.

The guide is organized in such a way that, after only one use, novice users will be familiar with most DMD Text/Graphics Package functions. The Text/Graphics Package software is easy to use. After gaining familiarity, the user should require only quick reference to the *5620 Dot-Mapped Display Reference Manual* manual pages for information.

* Registered trademark of AT&T Teletype Corporation

This guide is organized as follows:

- Chapter 1, ''INTRODUCTION,'' provides information about this guide and general information about the DMD Text/Graphics Package.
- Chapter 2, ''CIP GRAPHICS PROGRAM,'' gives detailed information for using the **cip** drawing program.
- Chapter 3, ''PROOF PHOTOTYPESETTER EMULATOR,'' gives details for using the phototypesetter emulator **proof** program.
- ''INDEX'' is an alphabetized subject listing with page numbers for easy subject location.

Text/Graphics Package Software

The Text/Graphics software in conjunction with *DOCUMENTER'S WORKBENCH** commands/programs allows you to enter, edit, format, and produce finished documentation for any purpose.

The UNIX† System commands **troff**, **nroff**, **pic**, and other text formatting support software are included in the DOCUMENTER'S WORKBENCH software add-on package. The DOCUMENTER'S WORKBENCH software provides a set of text processing commands for easy, flexible, and professional documentation.

> ***Note:*** To effectively use the Text/Graphics Package (**cip** and **proof**), your UNIX System must have the DOCUMENTER'S WORKBENCH software.

* Trademark of AT&T Technologies, Inc.

† Trademark of AT&T Bell Laboratories

The DMD Text/Graphics Package programs access and interface with the UNIX Operating System **nroff** and **troff** text formatters. Drawings can be easily developed on the DMD using **cip** and included with the text to be formatted by the **troff** formatter on a phototypesetter using **pic**. The formatted **troff** output can be reviewed on the DMD screen using **proof** to check for errors and format details (without generating paper). This process saves paper and the costs associated with the paper generation.

DMD Users

Users of the Text/Graphics Package software do not require any special computer skills or knowledge. However, users should first become familiar with basic operation of the DMD terminal and the Core software package. If you are not familiar with the DMD terminal, read or review the *5620 Dot-Mapped Display User Guide* which was included with your terminal.

Chapter 2

CIP GRAPHICS PROGRAM

PAGE

Chapter 2

CIP GRAPHICS PROGRAM

INTRODUCTION

The DMD interactive picture drawing program is named **cip**. The **cip** program provides easily selectable template shapes for creating drawings and editing them in a layer. Pictures drawn in a **cip** layer can be saved in UNIX System files as **pic** descriptions, and be included with text, tables, and equations as input to **troff**.

The mouse provides positional information and uses the three buttons to trigger frequently used commands. The keyboard is used only for entering text strings and file names.

The **cip** program can describe and position circles, boxes (solid, dotted, and dashed), ellipses, lines (solid, dotted, dashed, with or without arrowheads), circular arcs, splines (with or without arrowheads), and text. In addition, there is a mechanism for treating a collection of drawn shapes as a single entity, hereafter called a macro.

> ***Note:*** All illustrations in this manual were drawn with the **cip** program.

THE CIP LAYER

Figure 2-1, depicts an active **cip** layer. The **cip** layer is divided into the following areas:

- The template of selectable shapes at the top of the layer.
- The middle and major picture area where drawings are constructed.
- The **Mouse Buttons** box which displays the current functions of the mouse buttons.
- The **Number of bytes remaining** box which displays the number of bytes remaining in memory. Font loading information and error messages also appear in this box.
- The box showing edit depth appears in the bottom center area whenever a macro is edited.

The mouse button functions are shown at the bottom of the **cip** layer for two reasons. First, they make it easy to get acclimated to using the mouse as the primary input device. Second, they help users determine the current state of the mouse buttons.

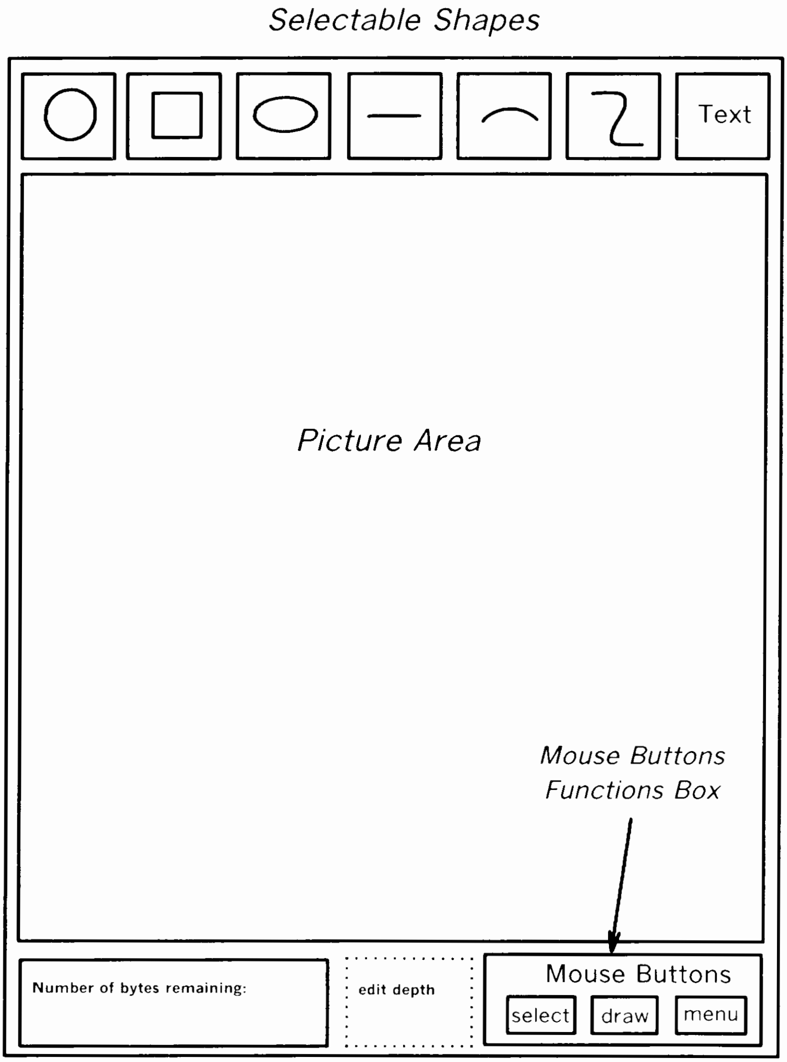


Figure 2-1. The Cip Layer

GETTING STARTED

The graphics **cip** program runs only in the **layers** environment. To load the **cip** program, first create a layer of any convenient size. Then type:

cip<RETURN>

The **cip** program will now begin downloading.

> ***Note:*** If the message **32ld: no memory in DMD** is displayed, too many layers are open and using up screen memory. Delete some of the layers or reduce their sizes and try again. Without the 1-megabyte memory option, it is impossible to load **cip** and **proof** at the same time.

When downloading is completed, depress button 3 and select **Large layer** from the menu that appears, then release button 3. The small **cip** layer will then expand to cover the entire screen.

Read the rest of this chapter and follow along on your DMD terminal; you will soon discover the power and simplicity of the **cip** graphics program.

Using the Mouse

IMPORTANT - Remember that the mouse buttons do not function like those of a keyboard. Depressing and releasing a mouse button are two distinct events, and the **cip** program makes use of this distinction. Menus are an example of this. Depressing a menu button will cause a menu to appear. The menu selection is made, however, when the button is released. Between these two events, the possible menu selection will track the cursor.

> ***Note 1:*** The mouse buttons are referred to as button 1, button 2, and button 3 (left to right, respectively) for a right-hand mouse.
>
> ***Note 2:*** "Depress" is used to describe the press and hold action and "release" means to let go of the mouse button. "Clicking" a button means "depressing" and "releasing" the button in quick succession.
>
> ***Note 3:*** Avoid moving the mouse while releasing button 2 or button 3 at the same time; an unwanted selection could be made. If button 2 or button 3 is depressed accidentally, remember to leave it down long enough to move the cursor outside the menu! Clicking button 1 cancels commands that require another click to execute after being selected.

Number of Bytes Remaining Box

This area of the cip layer is used for displaying the number of bytes of memory remaining in the DMD terminal. Each time an object is drawn, copied or deleted, this number is updated. If the number gets below about 200 bytes, the user should save the current picture. Alternatively, the user may be able to free up memory by deleting other layers.

If the error message:

Out of Storage: PUT and QUIT

is displayed, it will be necessary to free up memory before saving the picture. To do this, delete a few objects until the number of bytes remaining goes back above 200.

This box is also used to display font-loading information and error messages.

Selecting Template Shapes and Drawn Objects

Button 1 is used to either select a template shape from the top row of the layer or a previously drawn object in the picture area.

To select template shapes:

At the top of the layer, position the cursor inside a box surrounding the desired template shape and click button 1. The selected template shape will be highlighted in inverse video until something else is selected.

To select previously drawn objects:

Position the cursor within the bounding box (perimeter) of the desired object and click button 1. The selected object will vanish as long as button 1 is held down, reappearing when button 1 is released. This vanishing act alerts the user to which object has been selected. If the selected object is not the desired one, reposition the cursor and try again. An object stays selected until either another object or template shape is selected. Several attempts may be required to select an object in an area that is congested.

To unselect shapes or objects:

Move the cursor to a blank area of the layer, click button 1, and select nothing. Selecting another template shape or object also works. The unselect is unsuccessful if the middle mouse button (bottom of layer) function is either **draw** or empty and the third button is **menu**.

A selected object must be unselected (by selecting nothing) to invoke the command menu rather than an object menu. Menus (command and object) are discussed next.

MENUS

General

Button 3 is used to display menus of commands or object refinements. The following describes how menus work.

A ''pop-up'' menu display appears at the cursor location when button 3 is depressed. The menu display remains visible as long as button 3 is held down.

The menu is a list of words representing a command or transformation. When a menu is displayed in the layer and the cursor is over the menu, one of the entries is highlighted in reverse video. As the cursor moves up and down the menu, the highlighted entry changes. If the cursor is moved outside the menu, nothing is highlighted or selectable. The entry that is highlighted when button 3 is released is the selected one. If no entry is selected, nothing happens. This allows you to leave a menu without selecting an entry.

Command Menu

The command menu appears whenever no drawn object is selected and button 3 is depressed. The command menu contains entries that affect the entire picture area or some part of it, but not a specific single object. The following list describes each command menu entry and its effect when selected. Clicking button 1 or 2 cancels selected commands that require another click after being selected.

get file — Draws a named **pic** file in the picture area.

put file — Puts the picture area image into a **pic** description file for future selection, editing, or printing.

clear screen — Clears the picture area. When selected, the cursor will change to **RU SURE** as a reminder that whatever is in the picture area will be lost (unless it was put in a file). Clicking button 3 will clear the picture area; clicking button 1 or 2 cancels a **clear screen** command.

redraw screen — Redraws the drawing in the picture area. Use this command if object clarity is poor due to continuous editing or after a **get file** operation.

define macro — Allows a collection of drawn objects to be treated as a single object.

grid — Causes a grid of dots to be displayed in the picture area (drawings are constrained to grid points).

reverse video — Inverts the video sense of the layer.

quit — Exits the **cip** graphics program. When selected, the cursor changes to **RU SURE** as a reminder that **cip** must be reloaded after **quit** is invoked. Use **put file** to save drawings for future use. Clicking button 3 will quit the **cip** program; clicking button 1 or 2 will cancel the **quit**.

Object Menus

Object menu commands are tailored to the type of object selected. If a drawn object is selected with button 1, then button 3 is depressed, a menu of commands that in some way transforms the selected object is displayed. All objects can be copied and deleted.

For example, the object menus for circles and ellipses only have entries for copying and deleting. Arcs and splines can be reflected around either the **x** or **y** axis of their boundary. Splines can also have arrowheads attached to either or both ends. Lines can be made solid, dotted, dashed, reflected, or have an arrowhead attached to either or both ends (end point closest to the cursor). Boxes can have dotted, dashed, or solid boundaries.

Text can be made with roman, bold, or italic fonts. Point sizes can range from two to twenty-four picas. Text can also be justified to the left, right, or center. The text object menu will also display the attributes of selected text. Default text values are center justified, roman font, and 10 point size.

The next section explains drawing and editing.

DRAWING

There are two ways to draw objects in the picture area: by using template selection or copying existing drawings. Both methods are explained.

Drawing New Objects

First select a template shape from the top of the layer with button 1; then use button 2 to draw objects of the selected template shape in the picture area. The location and size of the drawn object are determined by the motions of the mouse and button 2.

For example, to draw a circle, select the circle template shape with button 1, move the cursor to the desired position for the center of the circle, and depress button 2. As long as button 2 is depressed, the trial radius of the circle will track the mouse movement. As the mouse is moved away from the origin, the circle will expand and vice versa. When button 2 is released, the circle radius is fixed as the distance between the origin and cursor position. Figure 2-2 defines the points fixed by button 2 motions for all template shapes.

> ***Note:*** The **cip** program draws objects using the XOR mode. This causes objects drawn over each other to erase parts of the other object. For example, if a circle is drawn and then a line is drawn which intersects the circle in two places, the two points of contact will appear blank. If the drawing is **troff**ed, the blanks will not appear.

Object	Start	Finish
circle	center	radius = distance
box	a corner	opposite corner

ellipse	center	height = abscissa (depress.y - release.y)
		height = abscissa (depress.x - release x)

line	endpoint	endpoint
arc	endpoint	endpoint center
spline	endpoint	point (subsequent releases define another point, until button 3 is clicked)
text	center of text string	a RETURN key ends text string

Figure 2-2. Points Fixed by Button 2 Motions

When an arc is drawn, it will be either one quarter or three quarters of a circle. If you want something else, select and edit the arc once it is drawn. Figure 2-3 shows the default arcs. The order that the endpoints are entered determines whether one-fourth or three fourths of a circle is drawn.

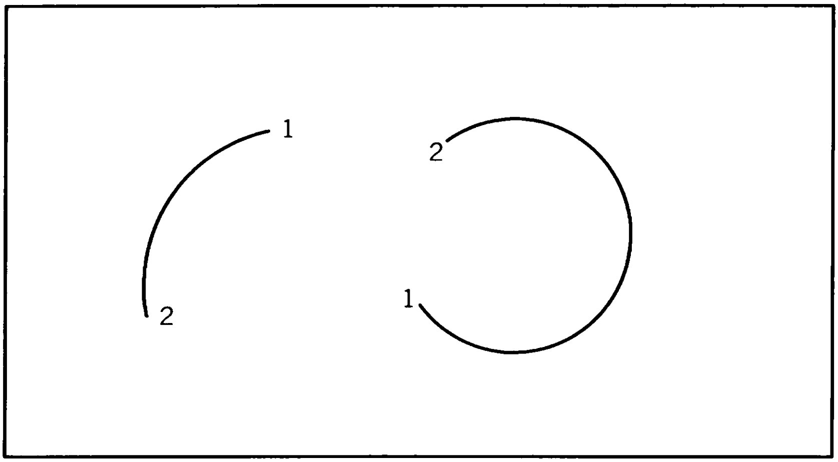
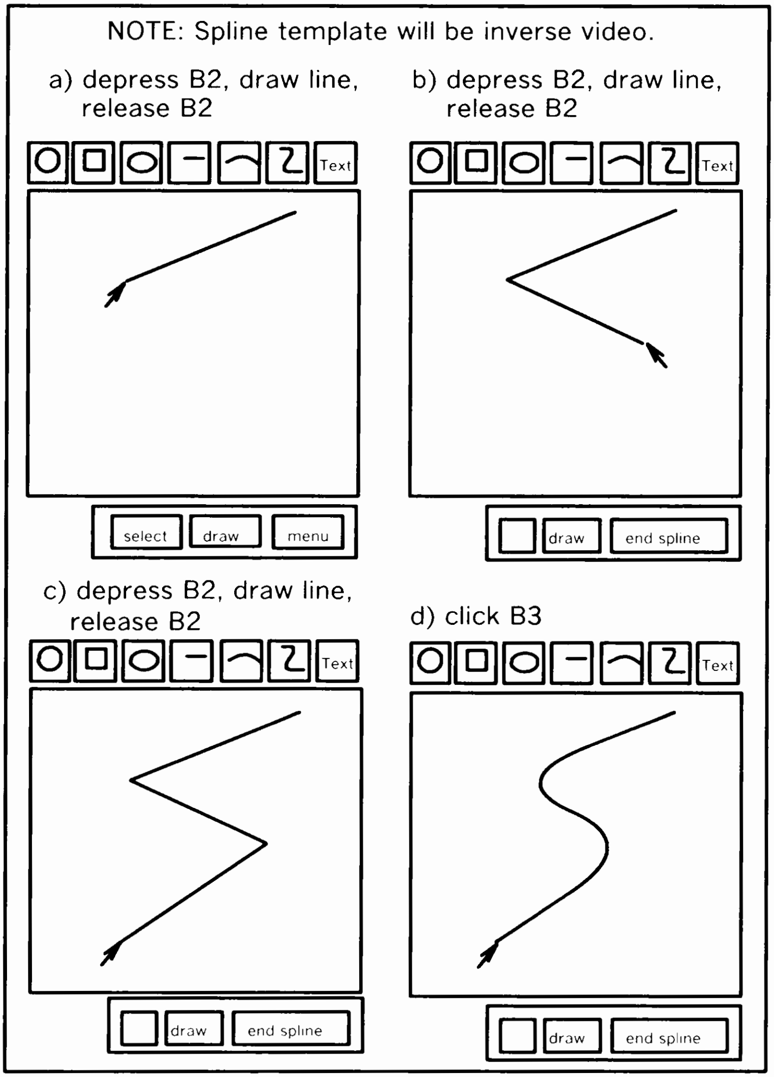


Figure 2-3. Default Arcs

Splines are curves that shape themselves to a framework of line segments. Each line segment is drawn by depressing button 2, drawing the line segment by moving the mouse, and then releasing button 2. Only the framework is shown while entering the spline. To terminate and draw the spline, click button 3. Figure 2-4 shows the process of drawing a spline.



NOTE: Spline template will be inverse video.

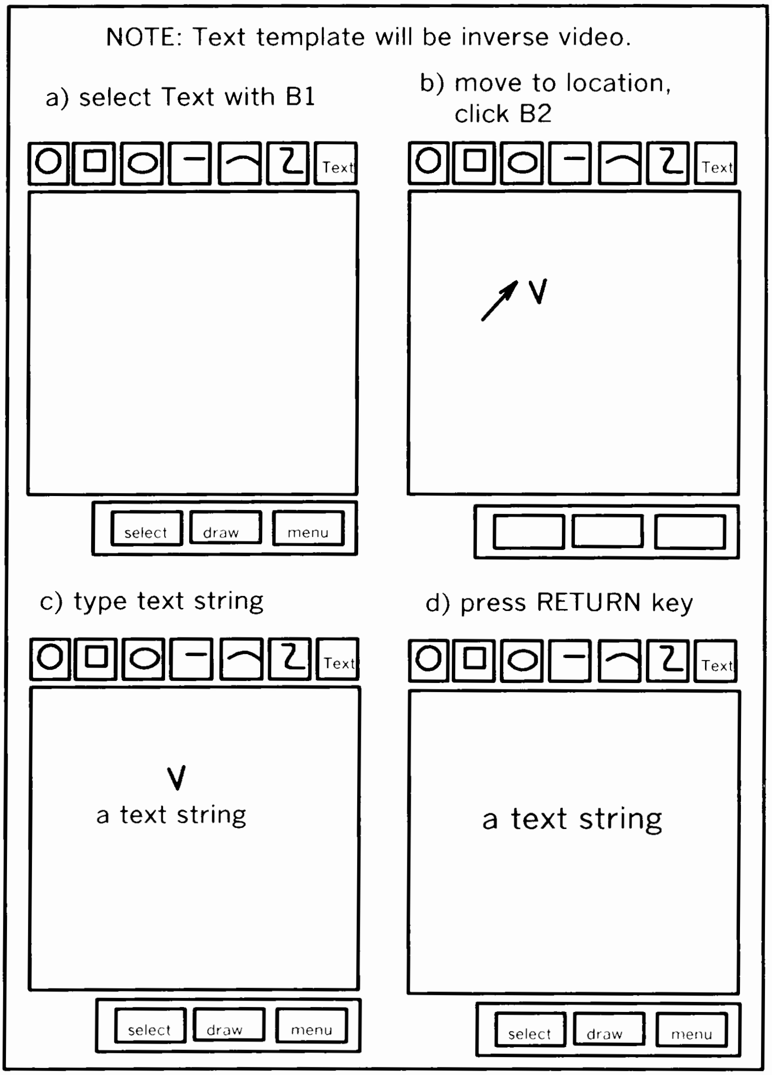
a) depress B2, draw line, release B2

b) depress B2, draw line, release B2

c) depress B2, draw line, release B2

d) click B3

Figure 2-4. Drawing a Spline



NOTE: Text template will be inverse video.

a) select Text with B1

b) move to location, click B2

c) type text string

d) press RETURN key

Figure 2-5. Adding Text

To include text in a drawing, first select the **text** template, move the cursor to the desired position for the center of the text string, and then click button 2. Now, type in the text. The backspace can be used to make minor corrections. CTRL W can be used to back up a word at a time and CTRL U can erase the entire line. As soon as, and not until, you hit the RETURN key, the string will begin to load in the picture area centered at the desired location. The point size and font type of the text being loaded will appear in the **Number of bytes remaining** box. Figure 2-5 shows the stages of adding text.

> ***Note:*** Default text values are center justified, roman font, and 10 point size.

To change the point size of text, first select the text string with button 1 and display the text object menu by depressing button 3. Select **point size** from the menu, then depress button 3 and a ''sub-menu'' of available point sizes (two to twenty-four picas) will appear. Select the desired point size from the ''sub-menu'' by releasing button 3. The **cip** program will now load the selected point size and then change the text string to that point size.

To change the font type or justification, select the desired entry from the object menu. The cursor changes to an hour glass when a text change command will take more than a few seconds.

Drawing boxes, lines, ellipses, and circles are straightforward. Simply select the desired template shape with button 1; then depress button 2 and move the mouse to draw the object. See Figure 2-6 for the origin of objects.

Object	Origin
circle	center
box	upper left corner
ellipse	center
line	first drawn point
arc	first drawn point
spline	first drawn point
text	center of text string
macro	upper left corner

Figure 2-6. The Origins of Objects

Copying Existing Objects

The second way to draw objects is to copy previously drawn objects. First, use button 1 to select a specific drawn object rather than a template shape. The selection is successful if the object disappears and then reappears. Then, depress button 3 to display a menu pertinent to the selected object. Select the menu entry **copy** by releasing button 3. Now, move the mouse to the desired location and click button 2 to position an exact copy of the selected object. The origin of the object (see Figure 2-6) will be located at the cursor position when button 2 is clicked.

Deleting Objects

To delete an object, first use button 1 to select the drawn object. Then, depress button 3 to display the object's menu. Select the object menu's first command entry **delete** by pointing at it and releasing button 3, and the selected object will be deleted.

Alignment Aids

The **cip** graphics program has the following features built in to make positioning objects easier:

- Lines that are drawn nearly horizontal or vertical will be made exactly so.

- Each of the basic objects has a number of sticky points. As you move the cursor near these special sticky points, it will tend to stick to them until the cursor is moved sufficiently far away. Figure 2-7 shows the sticky points for each template shape.

- The final alignment aid is **grid**. When the grid is turned on, drawn objects will ratchet from grid point to grid point as you move the cursor. The cursor favors sticky points; it will move to the nearest grid point only when there is no close sticky point. The grid is turned on and off with button 3.

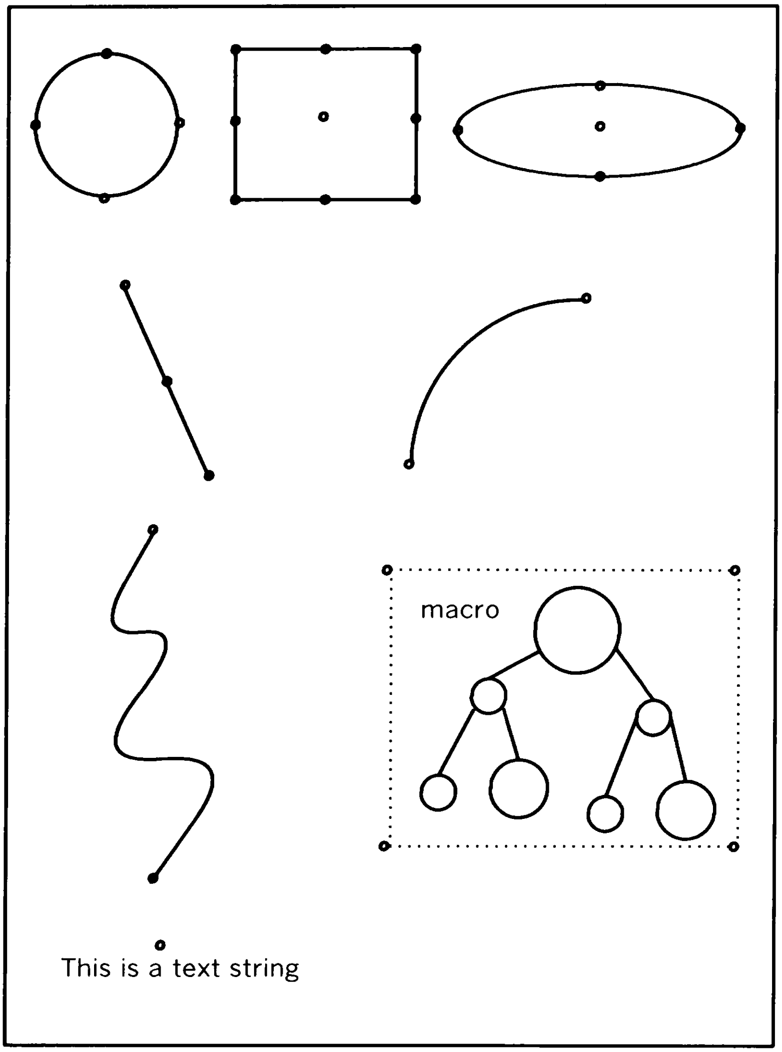


Figure 2-7. Sticky Points

Editing

Editing facilities allow the position and size of previously drawn objects to be changed. The basic method is to select the object with button 1, and then use button 2 for the actual editing. The position of the cursor, when button 2 is depressed, and the shape of the selected object determines the editing action (position or size change). See Figures 2-8 and 2-9.

For example, suppose a circle has been selected with button 1. If the cursor is near the circle origin (center) when button 2 is depressed, the circle origin will track the cursor as long as button 2 is depressed. Its new location will be the cursor position when button 2 is released, and the radius will be unchanged. If instead, the cursor is near the edge of the circle when button 2 is depressed, the circle origin will remain fixed and the radius will track the cursor, thus making the circle larger or smaller.

Rectangles and ellipses are edited in the same manner: near the center moves the object; near the edge (or corner, or endpoint) changes the size.

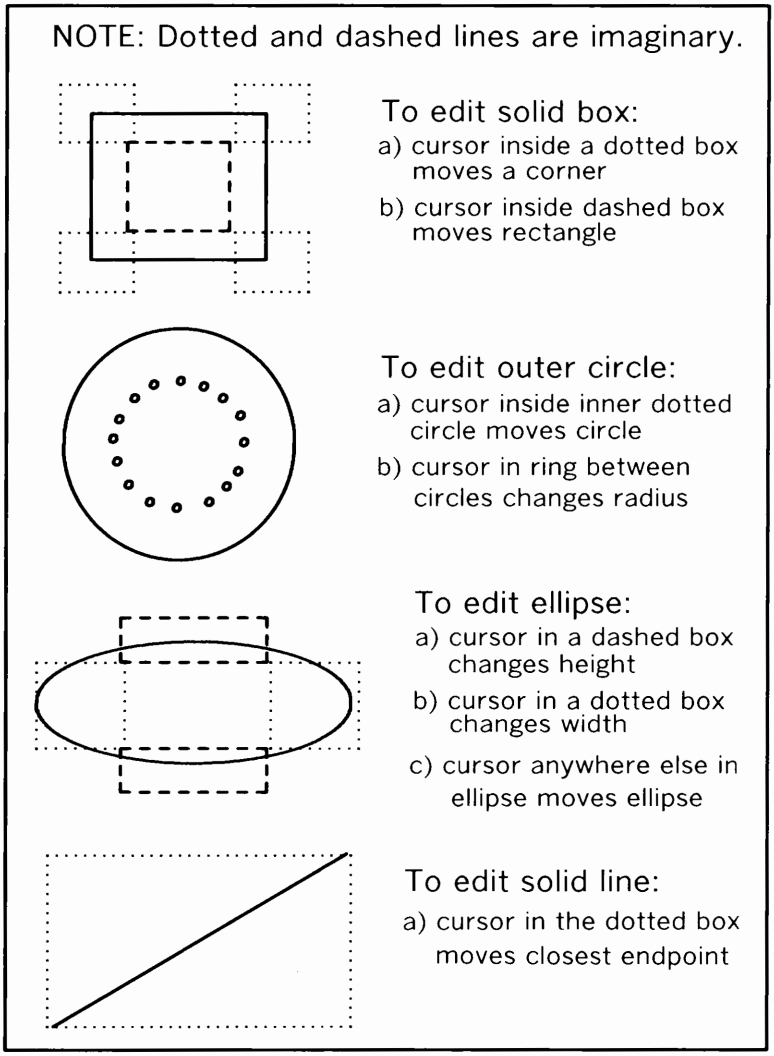
Lines can be moved one end at a time. The only other way to move a line is to make it a macro. Macros are discussed later.

Arcs have three interesting points: the two endpoints and the center of the circle that contains the arc. The point nearest the cursor when button 2 is depressed will track the cursor. This includes the center of the circle, which is shown with a pair of lines when the arc is selected. The only way to reposition an arc is to make the arc a macro.

Splines have a variable number of ''guiding points'' that can be changed. Lines between the guiding points are drawn to help in locating them when the spline is selected. The point nearest the cursor when button 2 is depressed will track the mouse motions, becoming fixed at the cursor position when button 2 is released. Again, there is no way to move a spline unless it is made into a macro.

Text can be moved by selecting the text string with button 1, then depressing button 2 and moving the cursor. The text will track the cursor, coming to rest when button 2 is released. The contents of a text string are changed by deleting the string and typing it in correctly.

Macro editing is slightly more complex. Changing the components of a macro is discussed next.



NOTE: Dotted and dashed lines are imaginary.

To edit solid box:

a) cursor inside a dotted box moves a corner

b) cursor inside dashed box moves rectangle

To edit outer circle:

a) cursor inside inner dotted circle moves circle

b) cursor in ring between circles changes radius

To edit ellipse:

a) cursor in a dashed box changes height

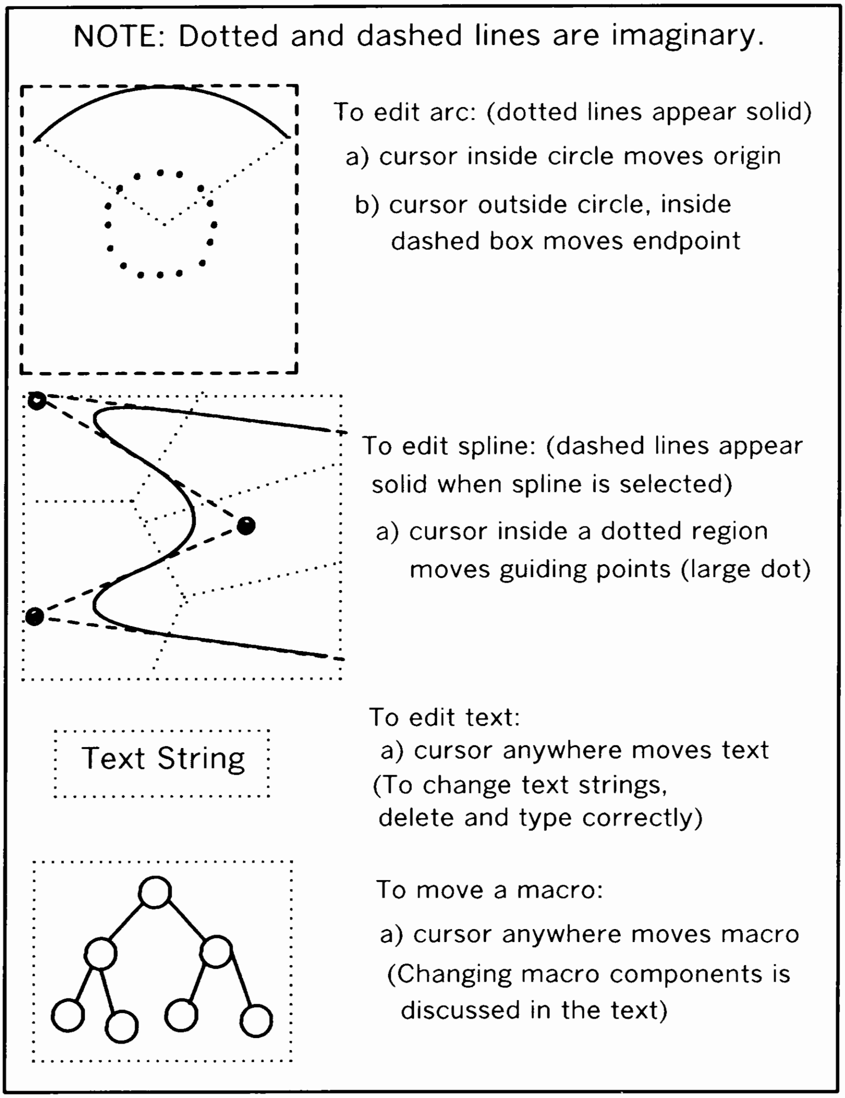
b) cursor in a dotted box changes width

c) cursor anywhere else in ellipse moves ellipse

To edit solid line:

a) cursor in the dotted box moves closest endpoint

Figure 2-8. Editing Circles, Boxes, Ellipses, and Lines



NOTE: Dotted and dashed lines are imaginary.

To edit arc: (dotted lines appear solid)

a) cursor inside circle moves origin

b) cursor outside circle, inside dashed box moves endpoint

To edit spline: (dashed lines appear solid when spline is selected)

a) cursor inside a dotted region moves guiding points (large dot)

To edit text:

a) cursor anywhere moves text (To change text strings, delete and type correctly)

To move a macro:

a) cursor anywhere moves macro (Changing macro components is discussed in the text)

Figure 2-9. Editing Arcs and Splines, Moving Text and Macros

Macros

The **cip** graphics program has a macro feature that allows objects to be grouped together and treated as a single object. A macro can be moved, reflected, copied, or deleted. In addition, components of a macro can be modified, deleted, or added. A group of objects is made into a macro by selecting the **define macro** option in the command menu. A bounding box is then drawn around the desired objects using button 2. Any object that is completely contained in the bounding box will be included in the macro. Macros can be "nested" (placing a macro(s) within a macro).

Once macro definition is complete, the macro instance can be selected with button 1, in the usual manner. The macro's bounding box will be highlighted in reverse video as long as button 1 is depressed. The macro is repositioned by selecting it with button 1, then depressing button 2 and moving the cursor. The macro's bounding box will track the cursor until button 2 is released. The macro object menu has entries for deleting, copying, reflecting, editing, and separating the macro instance.

> ***Note:*** After a macro has been reflected, the picture area is redrawn. This is due to the fact that when a macro is reflected, all occurrences of that macro are reflected. Redrawing occurs at the top level picture and not within macro editing (see next paragraph).

Selecting a macro's **edit** entry allows components to be modified, deleted, or added. Selection, drawing, editing, and menus work exactly as described before, except they now operate within the context of a macro. When macro editing begins, an **edit depth** box appears in the bottom center of the layer. It shows the macro's current "nested" editing depth; the count increases for each "nested" macro being edited. To terminate editing, move the cursor inside the **edit depth** box and click button 1, decrementing the depth count for each "nested" macro. When the count goes to zero, the picture area is redrawn to reflect the changes.

> ***Note:*** Modifications to a macro will affect all copies of the macro.

Detail can be added to a single macro instance by drawing over it, then making the macro a macro (''nesting''). Components of an individual macro cannot be deleted or modified without changing all copies.

Suppose you define a macro that contains a circle in a box. After replicating it several times, you decide that one copy should not have the circle. The macro menu entry, **separate**, will separate the macro back into individual objects. Now you can select the offending circle and delete it without changing the macro copies. Changes made to a macro will not affect separated objects.

> ***Note:*** If the **clear screen** command is selected while editing a macro the whole screen is cleared, but only the current macro is actually deleted. The screen is redrawn with all other drawings intact when **edit depth** is returned to zero (no macro editing).

Saving Pictures

The command menu has two entries that involve file operations. **Put file** converts the drawing in the picture area into a **pic** description file and saves it on the host computer.

Similarly, **get file** reads a *filename* presumed to be a **pic** description and draws the resulting file in the picture area. When **get file** is selected, the cursor changes to an hour glass until the file is completely read in. An empty box the size of the drawing's perimeter will appear. Now, keep the drawing centered or move it to another location. Click button 1 to keep the drawing centered. Depress button 2 to move the drawing; its location is defined when the button is released.

> ***Note:*** More than one drawing file can be loaded with **get file**. The only limit is the amount of memory available.

The prompt **FILE** appears for both commands. The *filename* from the previous **get file** or **put file** command will be automatically displayed to reduce the amount of typing necessary. A RETURN uses the default *filename*. Use BACK SPACE, CTRL W or CTRL U to erase the default *filename* and type in the desired *filename* if the default *filename* is not to be used.

There are four error conditions that can arise following a file operation. Attempts to **put file** are the least error prone: the only error, besides the host computer being down, is that you do not have permission to write the file in question. **Get file** commands are subject to errors involving access rights to the file, problems opening a pipe to parse the **pic**, and having a picture that is too complicated to fit within the available memory of the DMD. Errors are reported in the lower left portion of the layer just below the picture area.

Combining Drawings and Documents

Drawings you create with the **cip** program can be included in **troff**ed (phototypeset) documents. First, use the **put file** command to put the drawing into a file. A **pic** description of the drawing will now be in your chosen file. Next, convert your file to a **troff** input file by using the **pic** command. The **troff** input graphics file can then be inserted into another **troff** document at the desired location. Finally, follow your particular location's method of obtaining **troff** output hard copies.

See the next page for a sample method that uses **pic**, **troff** and **proof** to format and view a document containing **cip** drawings (**pic** descriptions).

Also see the **cip** manual page in the *5620 Dot-Mapped Display Reference Manual.*

Typesetting Drawings

Phototypeset documents and/or **cip** drawings can be examined with the **proof** program (usually done before obtaining hard copies). Here is a simple example of how to take a **cip** drawing, combine it with a **troff** document, and display the document on the screen by using the **proof** program. With slight modifications, this example can also be used to obtain hard copies.

1. While in **cip**, save your drawn image with the **put file** command. We will use the file name *cip.out*.

2. Exit or reshape the **cip** layer.

   Note: Without the 1-megabyte memory option, exit **cip** (since it is impossible to load **cip** and **proof** at the same time).

3. Run the UNIX System **pic** preprocessor as follows:

   pic *[options]* **cip.out** > **troff.in**

   Note: At this point, the file *troff.in* could be inserted in a **troff** document input file.

4. **Troff** the *troff.in* file as follows:

   troff *[options]* **troff.in** > **proof.in**

5. Display the output in a layer with the **proof** program, as follows:

   proof proof.in

See Chapter 3 for details on using the **proof** program.

Chapter 3

PROOF PHOTOTYPESETTER SIMULATOR

Chapter 3

PROOF
PHOTOTYPESETTER SIMULATOR

INTRODUCTION

The **troff** phototypesetter simulator program is named **proof**. The **proof** program allows you to use the DMD as a **troff** output simulator. As a simulator, formatted **troff**ed documents are viewed on the screen as if they were phototypeset. This provides the advantage of being able to review and check documents for errors and format details without costly generation of hard copies. Another important advantage of **proof** is that while reviewing the document in one layer, you can make any necessary corrections in a different layer.

The **proof** program runs only in the **layers** environment. **Proof** is simple to use, but first you must have a document (file) that has been formatted by **troff**.

USING PROOF

There are two ways to preview a given **troff** output file with **proof**: window or scale mode. The scale mode (-s option) means that each page of text reviewed will be compressed into the size of the layer. If the layer is made relatively small, you may not be able to read it but you can see what it looks like. The window mode (-w option) displays the text at the correct size, but only what the layer can hold is displayed. If neither option is selected, **proof** defaults to the scale mode.

> ***Note:*** The **proof** program can be downloaded into more than one layer.

***To use* proof**:

1. Download the **layers** program.

   > ***Note:*** Using the entire screen is not required but it gives a better looking phototypeset simulation. However, the -s or -w option can be used if the layer is much smaller than full screen size.

2. Create a layer.

3. At the keyboard, type:

   proof [-s -w] *filename*<RETURN>

   (where *filename* is the name of a **troff**ed file to be displayed). The layer will begin downloading the **proof** program. The cursor changes to a coffee cup, and the layer begins to fill with inverse video.

4. Shortly after downloading is completed, your simulated phototypeset file output will be displayed in the layer.

The **proof** program displays the name and size of the font it is trying to load before drawing the text. If unable to load that font, it will display an error message, use the default font, and continue.

After a layer's worth of text is displayed, **proof** pauses for a command and the mouse cursor is displayed as **CMD** to indicate commands will now be accepted. Commands can be made from the keyboard or with the mouse.

To control **proof** from the keyboard, use the following commands:

RETURN	Turns the page.
p *n*	Sets the next page to be printed to *n*. **$** selects the last page.
s	Compresses each page of text into the size of the layer (same as -s option). Turns the page.
w	Changes text size to display correct text size (same as -w option). Turns the page.
q	Quits but leaves a proof terminal.
x	Exits and restarts the regular layer terminal program. Works when **proof** is in the command mode.
unproof	Exits and restarts the regular layer terminal program, when not in the command mode.

To control **proof** with mouse, use the following button 3 menu commands:

next Turns the page.

page Sets the next page to be printed to *n*. **$** selects the last page.

scale Compresses each page of text into the size of the layer (same as -s option). Turns the page.

window Changes text size to display correct text size (same as -w option). Turns the page.

quit Quits but leaves a proof terminal.

exit Exits and restarts the regular layer terminal program.

Note: The default exit (end of file) from **proof** leaves a ''**proof** terminal'' which you can later use without reloading **proof**. The **proof** layer almost completely emulates a regular layer (escape sequences are not supported in a **proof** layer).

FONT SUBSTITUTION

When a particular size font is unavailable, **proof** substitutes the standard terminal font (or nothing, in the case of special characters). Up to fifty font types can be loaded at any one time. If this limit is exceeded, **proof** will still work correctly, but may run significantly slower.

Also, the **proof** program uses screen memory to store the fonts that it uses. If this memory is used up, a message to this effect will appear in the upper left-hand corner, and the default font will be substituted.

Index

Your comments and suggestions are appreciated and will help us to provide the best documentation for your use.

1. How would you rate this document for COMPLETENESS? (Please Circle)

Excellent		Adequate		Poor
4	3	2	1	0

2. Identify any information that you feel should be included or removed.

____

____

3. How would you rate this document for ACCURACY of information? (Please Circle)

Excellent		Adequate		Poor
4	3	2	1	0

4. Specify page and nature of any error(s) found in this document.

____

____

5. How would you rate this document for ORGANIZATION of information? (Please Circle)

Excellent		Adequate		Poor
4	3	2	1	0

6. Describe any format or packaging problems you have experienced with this document.

____

____

7. Do you have any general comments or suggestions regarding this document?

____

____

8. We would like to know a little about your background as a user of this document:

A. Your job function ____________________

B. Number of years experience with computer hardware: operation ,
maintenance .

C. Number of years experience with computer software: user ,
programmer .

Your Name Phone No. ____________

Company ______________________________

Address ______________________________

City & State Zip Code ______

AT&T

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 1999 GREENSBORO N C

POSTAGE WILL BE PAID BY ADDRESSEE

DOCUMENTATION SERVICES
2400 Reynolda Road
Winston-Salem, N.C. 27106-9989

Do Not Tear—Fold Here and Tape

COMMENT FORM

Your comments and suggestions are appreciated and will help us to provide the best documentation for your use.

1. How would you rate this document for COMPLETENESS? (Please Circle)

   Excellent Adequate Poor
   4 ------------------------3 ------------------------2 ------------------------1 ------------------------0

2. Identify any information that you feel should be included or removed.

   ____

   ____

3. How would you rate this document for ACCURACY of information? (Please Circle)

   Excellent Adequate Poor
   4 ------------------------3 ------------------------2 ------------------------1 ------------------------0

4. Specify page and nature of any error(s) found in this document.

   ____

   ____

5. How would you rate this document for ORGANIZATION of information? (Please Circle)

   Excellent Adequate Poor
   4 ------------------------3 ------------------------2 ------------------------1 ------------------------0

6. Describe any format or packaging problems you have experienced with this document.

   ____

   ____

7. Do you have any general comments or suggestions regarding this document?

   ____

   ____

8. We would like to know a little about your background as a user of this document:

A. Your job function ____________________

B. Number of years experience with computer hardware: operation .
maintenance .

C. Number of years experience with computer software: user .
programmer .

Your Name Phone No. ____________
Company ____
Address ____
City & State Zip Code ______

AT&T

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 1999 GREENSBORO, N C

POSTAGE WILL BE PAID BY ADDRESSEE

DOCUMENTATION SERVICES
2400 Reynolda Road
Winston-Salem, N.C. 27106-9989

Do Not Tear—Fold Here and Tape