15. DISPLAYSTREAMS

A displaystream is a generaJized "place to display". They determine exactly what is displayed where. One example of a displaystream is a window. Windows are the only displaystreams that will be used in this chapter. If you want to draw on a bitmap that is not a window, other than with BITBLT, or want to use other types of displaystreams, please refer to Chapter 27 in the *IRM*.

This chapter explains functions for drawing on displaystreams: DRAWLINE, DRAWTO, DRAWCIRCLE., and FILLCIRCLE. In addition, functions for locating and changIng your curreAt position in the displaystream are covered: DSPXPOSITION, DSPYPOSITION, and MOVETO.

Drawing on a Displaystream

The examples belowshow you how the functions for drawing on a display stream work. First, create a window. Windows are displaystreams, and the one you create are used for the examples in this chapter. Type:

```
(SETQ EXAMPLE.WINDOW (CREATEW))
```

DRAWLINE

DRAWLINE draws a line in a displaystream. For example, type:

```
(DRAWLINE 10 15 100 150 5 'INVERT EXAMPLE.WINDOW)
```

The results should look like Figure 15-1:

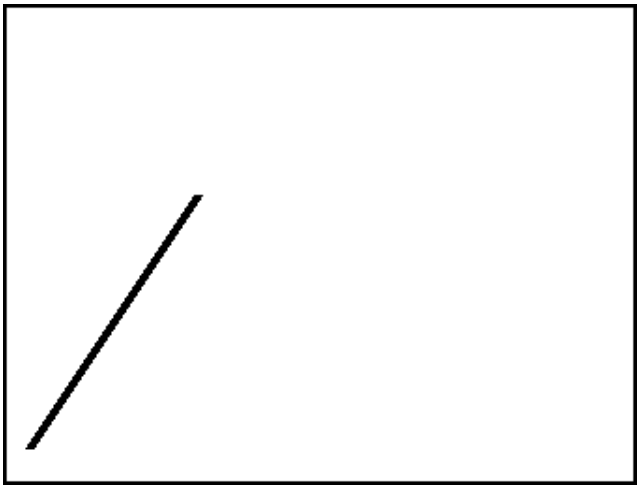

Figure 15-1. Line Drawn onto the EXAMPLE.WINDOW Displayrtream

The syntax of DRAWLINE is

(DRAWLINE *x1 y1 x2 y2 width operation stream color dashing*)

The coordinates of the left bottom corner of the displaystream are 0 0.

xl and yl	x and y coordinates of the beginning of the line
x2andy2	ending coordinates of the line
width	width of the line, in pixels
operation	way the line is to be drawn. `INVERT` causes the line to invert the bits that are already in the displaystream. Drawing a line the second time using `INVERT` erases the line. For other operations, see Chapter 27 in the *IRM*.
stream	displaystream. In this case, you used a window.

DRAWTO

`DRAWTO` draws a line that begins at your current position in the displaystream. For example, type:

```
(DRAWTO 120 135 5 'INVERT EXAMPLE.WINDOW)
```

The results should look like Figure 15-2:

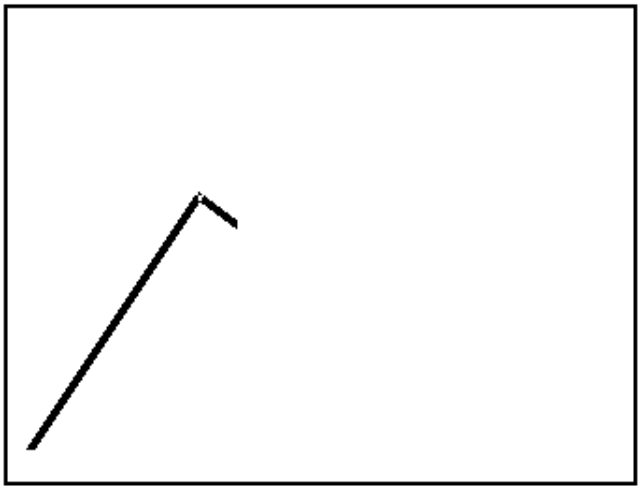

Figure 15-2. Another Line drawn onto the `EXAMPLE.WINDOW` Displaystream

The syntax of `DRAWTO` is

(`DRAWTO` *x y width operation stream color dashing*)

The line begins at the current position in the displaystream.

x	x coordinate of the end of the line
y	y coordinate of the end of the line
width	width of the line
operation	way the lino is to be drawn. `INVERT` causes the line to invert the bits that aro already in tho displaystream. Drawing a line the second time using `INVERT` erases the line. For other operations, see Chapter 27 in the *IRM*
stream	displaystreom. In this case. you used a window.

DRAWCIRCLE

DRAWCIRCLE draws a circle on a displaystream. To use it, type:

```
(DRAWCIRCLE 150 100 30 '(VERTICAL 5) NIL EXAMPLE.WINDOW)
```

Now your window, EXAMPLE.WlNDOW, should look like Figure 15-3:

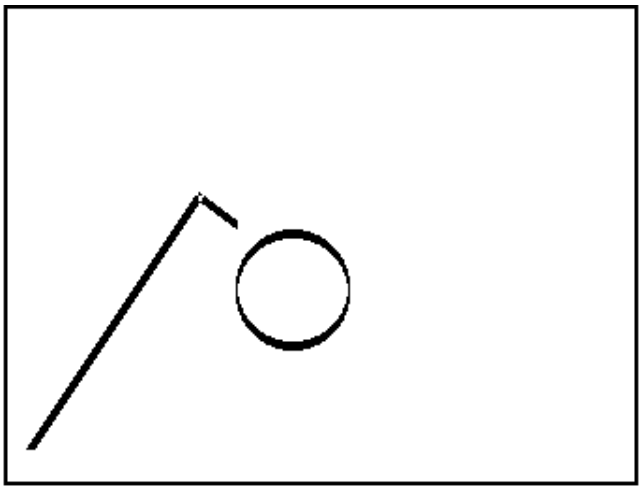

Figure 15-3. Circle Drawn onto the EXAMPLE.WlNDOW Displaystream

The syntax of DRAWCIRCLE is

(DRAWCIRCLE *centerx centery radius brush dashing stream*)

centerx	x coordinate of the center of the circle
centery	coordinate of the center of the circle
radius	radius of the circle in pixels
brush	list.- The first- item of the list is the shape of the brush. Some of your options include ROUND, SQUARE, and VERTICAL. The second item of that list is the width of the brush in pixels.
dashing	list of positive integers. The brush is "on" for the number of units indicated by the first element of the list, "off" for the number of units indicated by the second element of the list. The third element specifies how long it will be on again, and so forth. The sequence is repeated until the circle has been drawn.
stream	displaystream. In this case, you used a window.

FILLCIRCLE

FILLCIRCLE draws a filled circle on a displaystream. To use it, type:

```
(FILLCIRCLE 200 150 10 GRAYSHADE EXAMPLE.WINDOW)
```

EXAMPLE.WlNDOW now looks like Figure 15-4:

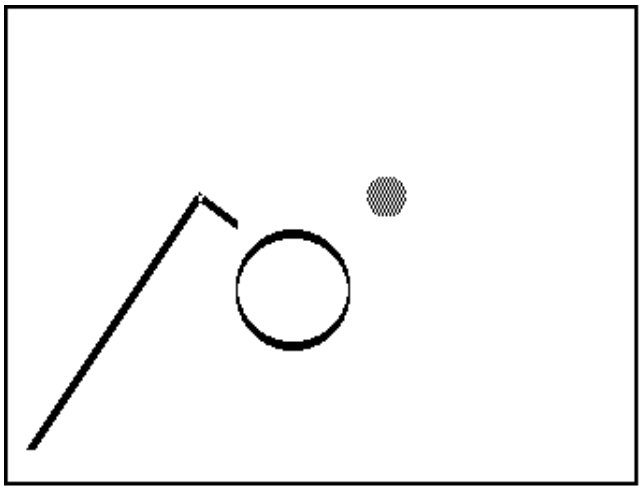

Figure 15-4. A filled circle drawn onto the displaystream

The syntax of FILLCIRCLE is:

(FILLCIRCLE *centerx centery radius texture stream*)

centerx	x coordinate of the center of the circle
centery	y coordinate of the center of the ci rcle
radius	radius of the circle in pixels
texture	shade that will be used to fill in the circle. Interlisp-D provides you with three shades: WHlTESHADE, BLACKSHADE, and GRAYSHADE. You can also create your own shades. For more information on how to do this, see Chapter 27 in the *IRM*.
stream	displaystream. In this case, you used a window

There are many other functions for drawing on a displaystream. Please refer to Chapter 27 in the *IRM*.

Text can also be placed into displaystreams. To do this, use printing functions such as PRIN1 and PRIN2, but supply the name of the displaystream as the "file" to print to. To place the text in the proper position in the displaystream, see the section below.

Locating and Changing Your Position in a Displaystream

There are functions provided to locate, and to change your current position in a displayitream. This can help you place text, and other images where you want them in a displaystream. This primer will only discuss three of these. There are others, and they can be found in the Chapter 27 of the *IRM*.

DSPXPOSITION

DSPXPOSITION is a function that will either change the current x position in a displaystream, or simply report it. To have the function report the current x position in EXAMPLE.WlNDOW, type:

```
(DSPXPOSITION NIL EXAMPLE.WINDOW)
```

DSPXPOSITION expects two arguments. The first is the new x position. If this argument is NIL, the current position is not changed, merely reported. The second argument is the displaystream.

DSPYPOSITION

DSPYPOSITION is an analogous function, but It changes or reports the current y position in a displaystream. As with DSPXPOSlTlON, if the first argument is a number, the current y position will be changed to that position. If it is NIL, the current position is simply reported. To have the function report the current y position in EXAMPLE.WlNDOW, type:

```
(DSPYPOSITION NIL EXAMPLE.WlNDOW)
```

MOVETO

The function MOVETO always changes your position in the displaystream. It expects three arguments:

(MOVETO *x y stream*)

x	new x position in the display stream
y	new y position in the display stream
stream	display stream. The examples so far have used a window