NAME

x11 – spec graphics under the X Window System

DESCRIPTION

You can have spec draw high resolution graphics in X Window System windows by selecting high resolution graphics (with the `setplot` macro) and setting the `GTERM` global variable to `x11`. You can set `GTERM` either as an environment variable in the shell or by assigning it the `"x11"` string while running spec. However, each time you start spec, the environment variable will override a previously assigned value to `GTERM`.

The process that spec creates (named *x11filt*) to do the plotting looks at the `DISPLAY` environment variable. The value of spec's built-in variable named `DISPLAY` will be propagated to the *x11filt* process environment when it is spawned. Thus it is possible to change the host on which the plots are drawn while running spec by assigning new values to `DISPLAY`. For example,

```
DISPLAY="host1:0"; plot_cntl("filter1,open")
DISPLAY="host2:0"; plot_cntl("filter2,open")
```

will create plot windows on two different hosts.

RESOURCE FILE ITEMS (*.Xdefaults)*

Some parameters associated with the X windows plot can be set in an *.Xdefaults* file in your home directory. You can include some or all of the following parameters:

```
spec.geometry:      =640x380-20+10
spec.Foreground:    black
spec.Background:    white
spec.Backstore:     on
spec.Retained:      off
spec.BorderWidth:   4
spec.Border:        black
spec.Font:          -*-*-r-*-*-*-*-*-*-*-*-*
spec.FontDebug:     0
spec.Colors:        on
spec.AutoRaise:     on
spec.DotSize:       small
```

The values for the parameters given above are the defaults.

Up to five X graphics windows may be used with spec simultaneously. Parameters specified in the *.Xdefaults* file with the keyword `spec` will apply to all, unless overridden by the following. Parameters specified with the keyword `spec_1`, will apply to the window selected with the command `plot_cntl("filter1")`, parameters specified with the keyword `spec_2` will apply to the window selected with the command `plot_cntl("filter2")`, etc. An additional *x11filt* process is created for each window.

The *geometry* parameter uses the conventional notation to set the width and height of the window and the offsets from the edges of the root window. All the values are in pixels.

The *Background* and *Foreground* parameters set the colors associated with color numbers zero and one, respectively.

The *Backstore* and *Retained* parameters select how the image in the window will be saved when the window is covered up by another window. The backing-store method can be the most efficient way to preserve the window contents if the graphics display supports it. For the retained-pixmap method, the *x11filt* process performs all the drawing operations on both the active display and an in-memory copy of the window, and uses the in-memory copy to redraw uncovered portions of the window on *expose* events. The retained-pixmap method is generally not as efficient as using backing store, but may be necessary if backing store is

unavailable. If backing store is requested, but unsupported by the display, the retained-pixmap method is automatically used.

The *BorderWidth* and *Border* parameters select the width and color of the window border, but are generally ignored when a window manager (such as *mwm*) is running.

The *Font* parameter uses the standard notation for specifying application fonts in X Windows. The default font uses the * to wild card all fields except the fourth, which selects a *roman* font. The graphics program selects among all the fonts that match the pattern the one that best fits the height and width of the window when the program first starts and each time the window is resized. The font selected can be limited to a single family in your *.Xdefaults* file with syntax as in following examples,

```
spec.Font: -*-lucidatypewriter-bold-r-*-*-*-*-*-*-*-*-*-*
spec.Font: -*-fixed-medium-r-*-*-*-*-*-*-*-*-*-*
spec.Font: -*-courier-medium-r-*
```

Note, your choice of a font can be aided with the X utility *xlsfonts* which lists all the available fonts, and the utility *xfontsel* which can be used to inspect the available fonts.

The *x11filt* program also supports *scalable* fonts introduced in X11R5, when such a font is selected using the *.Xdefaults* resources or the `plot_cntl() "font="` option described below. For scalable fonts, the server uses an algorithm to fit the font to whatever size is requested. Scalable fonts can be identified by the zeros in the 7th, 8th and 12th fields of their long name. For example,

```
spec.Font: -adobe-courier-bold-r-normal--0-0-0-0-m-0-iso8859-1
spec.Font: -adobe-courier-medium-r-normal--0-0-0-0-m-0-iso8859-1
```

are a couple of possible scalable fonts.

Please note that on some older systems, the *fonts.dir* files in the */usr/lib/X11/fonts* directories aren't up to date, so the *x11filt* graphics program will try to load a font that doesn't exist. This problem can be fixed by having the system administrator run the `mkfontdir` command.

Setting the *FontDebug* property to a nonzero value will cause the *x11filt* filter process to display some font diagnostics on the terminal screen, which may be useful if there are problems with the font appearance. Debugging can be turned off either by setting the property value back to zero or with the `plot_cntl()` 9900 command described below.

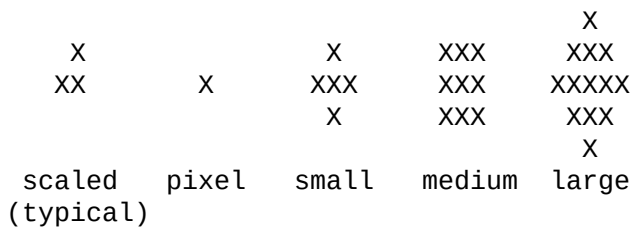
The program will use color if it determines you are using a color display. The *Colors* parameter lets you turn use of colors off.

By default, the graphics window will be brought to the top of the stacking order among its siblings each time its contents are changed. To disable this feature, set the *AutoRaise* parameter to off.

The *DotSize* parameter controls the size of the smallest point drawn by spec. Possible values are

```
scaled    pixel     small     medium    large
```

The pixel patterns are as follows:



The `scaled` points use the X library `XFillArc()` routine to draw a filled circle. The result depends on the platform and is often nonsymmetric. Before being a configurable option, the default dot size was `scaled`. The default is now `small`.

`plot_cntl()` **COMMANDS**

You can assign a title to the graphics window using the command

```
plot_cntl("title=This is a title")
```

from spec. The title can be changed at any time.

You can also change the size and position of the window at any time using the `plot_cntl()` function as in

```
plot_cntl("geometry=640x380-20+10")
```

Fonts can be selected using

```
plot_cntl("font=-adobe-courier-bold-r-normal--20-140-100-100-m-110-iso8859-1")
```

for example. If a non-scalable font is chosen this way, the font size will no longer change to match the size of the window. Any valid font name can be used, including short alias names, as in,

```
plot_cntl("font=8x16")
```

Several special features can be accessed using

```
plot_cntl("cmd=numb")
```

(as of spec release 5.01.02-7) where the following values for *numb* are recognized:

`9109` – turns auto-raise mode on
`9110` – turns auto-raise mode off
`9120` – selects scaled dot size
`9121` – selects single pixel dot size
`9122` – selects small dot size
`9123` – selects medium dot size
`9124` – selects large dot size
`9666` – creates a detached window (see below)
`9900` – turns font debugging off
`9901` – turns font debugging on

Also, the *numb* argument can be a space- or comma-separated list of numbers.

DETACHING A WINDOW

You can make a detached copy of an active spec X11 plot window by pressing the control key and any mouse button while the mouse cursor is over the window (as of spec release 5.01.02-7). A new X11 window will be created that contains the contents of the original window. The contents of the new window will no longer change, and the window will remain on the screen, even after spec exits. Use the window manager close button or menu item to delete the window.

SEE ALSO

`colors setplot splot`