CORONAL DIAGNOSTIC SPECTROMETER

SoHO

CDS SOFTWARE NOTE No. 5

Version 1.1 | 14 September 1994

SERTS GRAPHICS DEVICES ROUTINES

W. Thompson
Applied Research Corporation
NASA Goddard Space Flight Center
Laboratory for Astronomy and Solar Physics
Code 682.1
Greenbelt, MD 20771, USA

William.T.Thompson.1@gsfc.nasa.gov
pal::thompson

These routines form the part of the SERTS subroutine library pertaining to switching back and forth between graphics devices. The philosophy behind these routines is to be able to switch from one device to another, and to be in the same state as when last using that device. That way one could do something like the following:

1. Start in X-windows modes. Start a plot.
2. Switch to PostScript mode. Start a plot there as well.
3. Switch back to X-windows mode. Do an overplot.
4. Switch back to PostScript mode. Also do an overplot.

and have all the plots work out correctly. The same philosophy is extended to switching back and forth between windows on the same device, and between multiple plots in the same window.

1 Switching between graphics devices

The ability to switch back and forth between different graphics devices without losing the state information for each device is accomplished through the routine SETPLOT (without underscore). This routine is called in the same way as the standard IDL routine SET_PLOT (with underscore), e.g.

```
SETPLOT,'PS'
```

The state of each graphics device is stored in an internal common block for later retrieval. SETPLOT will only work properly if it is used exclusively to change between graphics devices.

There are a number of routines that serve as a front end to SETPLOT. These are

TEK	Tektronix terminal
REGIS	Regis terminal
SUNVIEW	SunView
XWIN	X-windows
WIN	Microsoft Windows
PS	PostScript plot file

Most of these do nothing more than call SETPLOT and print a simple message to the screen. However, the PS routine provides some additional capabilities described below—one should use the command PS instead of SET_PLOT,'PS'.

Certain system variables that control the appearance of the plot can be defined to be different depending on the graphics device. These include

!P.CHARSIZE
!P.FONT
!P.COLOR
!P.BACKGROUND

This means that users should be aware that changing the values for one device does not automatically change it for another device. Thus, for example, if one is making plots both to an X-windows display, and to a PostScript file, and wants to change the character size on both displays, the system variable !P.CHARSIZE has to be changed twice, e.g.

```
!P.CHARSIZE = 2         ;Generate plot on X-windows display.
PLOT, A

PS                      ;Generate plot in PostScript file.
!P.CHARSIZE = 2
PLOT, A

XWIN                    ;Switch back to X-windows.
```

Once they have been set, however, SETPLOT will remember the settings individually for each graphics device.

As well as switching between graphics devices, SETPLOT is also used to define default values for several system variables (e.g. !P.FONT) as a function of graphics device. Users installing this software on their own system can modify this section of the code to suit their own needs.

2 Switching between windows

The routine SETWINDOW can be used to switch between two or more already existing windows in the same way that SETPLOT switches between devices. Like SETPLOT, one can switch back and forth between two separate graphs in each window. It is called in the same way as WSET, e.g.

```
SETWINDOW, 3
```

Creating a new window with the WINDOW command will also automatically switch to that window. In order to save the settings for the current window, it may be necessary to call SETWINDOW before calling WINDOW, e.g.

```
SETWINDOW
WINDOW, 2
```

(Calling SETWINDOW without any parameters saves the settings for the current window.)

3 Switching between multiple plots on one screen

One can also place multiple plots on the same screen, and switch between them, using the routine SETVIEW. The results are similar to using !P.MULTI, but the process is different. The major advantage of using SETVIEW instead of !P.MULTI is the ability to switch back and forth between the various plots, and still be able to use commands such as OPLOT, even if other plots have been generated in between.

The command

```
SETVIEW, 1, 3, 2, 5
```

means that the next plot will be one in a series of plots—specifically, it will be the first of three from the left, and the second of five from the top.

As an example, suppose that one wants to make three plots in a row of the quantities A, B, and C. You can accomplish this by doing something like the following:

```
ERASE
SETVIEW, 1, 3
PLOT, A
SETVIEW, 2, 3
PLOT, B
SETVIEW, 3, 3
PLOT, C
```

Calling SETVIEW with non-trivial parameters, as shown in the example, automatically disables the automatic erase ordinary generated by commands such as PLOT. Thus, the ERASE command is needed to ensure that one starts with a blank screen. The plots are shown as being made in order from left to right, but actually could be produced in any order.

Once the plots have been generated as shown, the quantity B_PRIME could be overlaid on top of the plot of B through the commands

```
SETVIEW, 2, 3
OPLOT, B_PRIME
```

Calling SETVIEW without any parameters resets the graphics behavior back to the default.

When switching between graphics devices, the view must be defined separately for each graphics device, but the view will be preserved when switching back and forth. For example,

```
SETVIEW, 1, 3, 1, 5     ;Generate plot on X-windows display.
PLOT, A

PS                      ;Generate plot in PostScript file.
SETVIEW, 1, 3, 1, 5
PLOT, A

XWIN                    ;Switch back to X-windows.
```

The same holds true for switching between windows using SETWINDOW.

4 PostScript support

There are several routines specifically oriented toward generating and processing PostScript output files. These are

PS	Open a PostScript file, or switch to an already opened file.
PSCLOSE	Close a PostScript file.
PSPLOT	Close a PostScript file, and send it to the printer.

4.1 Opening PostScript files

The routine PS can be used to redirect graphics output to a PostScript file. The name of the file can either be passed explicitly, e.g.

```
PS, 'myfile'
```

or the filename can be omitted and then the default name "idl.ps" is used. If the filename is passed without any extension, as in the above example, then the extension ".ps" is used. Once the PostScript file is opened, subsequent calls to PS (without any parameters) returns to the previously opened file, e.g.

```
PLOT, A                 ;Plot on X-windows display

PS, 'myfile'            ;Open PostScript file myfile.ps
PLOT, A

XWIN                    ;Return to X-windows display
OPLOT, B

PS                      ;Return to myfile.ps
OPLOT, B
```

The default orientation for plots produced with the PS command is landscape mode—i.e. the plot is oriented with the X-axis aligned with the long dimension of the paper. If portrait mode is desired instead, then the keyword /PORTRAIT should be added to the command, e.g.

```
PS, /PORTRAIT
```

In this case, portrait mode means that not only is the X-axis aligned with the short dimension of the paper, but also that the plot will be taller than it is wide. Using the PS routine, both the landscape and portrait orientations make use of the entire page. The /LANDSCAPE keyword is the inverse of /PORTRAIT.

Encapsulated PostScript files, as used in incorporating into TEX and LATEX documents (and in other software) can be generated by using the /ENCAPSULATED keyword, e.g.

```
PS, /ENCAPSULATED, 'myfile.eps'
```

An explicit filename is always required with the /ENCAPSULATED switch.

Color PostScript files can be generated with the /COLOR switch. This can be combined with any of the other switches already discussed: /LANDSCAPE, /PORTRAIT, or /ENCAPSULATED.

Also, the COPY switch can be used to copy the current color tables into the PostScript file. Since a call to SETFLAG is generated, the SERTS image display software is required to use the COPY switch.

It should be noted that calling PS with any of the above keywords, or with a filename, will force IDL to open a new PostScript file regardless of whether there's already one open or not.

4.2 Closing and printing PostScript files

PostScript files can be closed with the command

```
PSCLOSE
```

This will not only close the PostScript file, but will also automatically redirect graphics output back to whatever device the user was using previously. Alternately, the command

```
PSPLOT
```

will not only close the file, but send it to the printer as well.

There are several methods that PSPLOT uses to decide which printer to use. One way is to specify the queue directly through the QUEUE keyword, e.g.

```
PSPLOT, QUEUE='myprinter'
```

If that's not passed, then PSPLOT checks for the existence of the system environment variable (VMS: logical name) PSLASER. Finally, on Unix systems the default print queue will be used if no other information is given—it is assumed that this is almost always a PostScript printer. However, on VMS systems the print queue has to be explicitly defined.

When printing color PostScript files, the environment variable PSCOLOR is used instead of PSLASER.

The /DELETE keyword tells PSPLOT to automatically delete the file. Depending on the operating system, this may occur either after the file has been queued for printing, or not until it has actually been printed.

If PSPLOT is used to close the file as well as print it, then it "remembers" what the name of the file is, and whether it was opened as a color PostScript file or not. However, once the file is closed, then it "forgets" this information, and reverts to the defaults (i.e., filename "idl.ps" and without color). In that case, this information would have to be passed explicitly, e.g.

```
PSPLOT, 'myfile.ps'
PSPLOT, 'mycolorfile.ps', /COLOR
```

5 Equal X and Y scales

The routine SETSCALE can be used to force the X and Y axes of a plot to be the same. In other words, one can use SETSCALE to ensure that squares will appear as squares, and circles will appear as circles, and not as rectangles and ellipses.

For example, suppose that the arrays X and Y represent positions in some cartesian coordinate system, and that they both have the same units (e.g. kilometers). The commands

```
SETSCALE, X, Y
PLOT, X, Y, PSYM=1
```

will plot the data so that the scale (e.g. kilometers/inch) will be the same in both directions. The corners of the plot will also be adjusted to fit the range of the data unless the keyword /NOADJUST is used.

An optional way to use SETSCALE is to pass it the ranges explicitly. For example:

```
SETSCALE, XMIN, XMAX, YMIN, YMAX
```

Or, if one's doing a contour plot, then one can pass the array to be contoured to SETSCALE:

```
SETSCALE, ARRAY
CONTOUR, ARRAY
```

Although if CONTOUR is to be called with the optional arrays for the coordinates along the X and Y axes, then those arrays should be passed to SETSCALE instead of the array to be contoured:

```
SETSCALE, X, Y
CONTOUR, ARRAY, X, Y
```

Calling SETSCALE by itself will reset the graphics behavior to the default. It's suggested that this be done right after generating the plot—it won't interfere with any overplotting commands. The routines discussed in the sections above, SETPLOT, SETWINDOW, and SETVIEW, will automatically do this before taking any actions.

The behavior of SETSCALE can be fine-tuned with the !ASPECT system variable. This variable describes the ratio of the height of a pixel over its width, normally 1. The SETPLOT routine will keep track of !ASPECT as a function of graphics device.