
Subject: Re: plotting graphics for web pages
Posted by [Paul Van Delst\[1\]](#) on Mon, 19 May 2008 21:41:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Paul van Delst writes:

>

>> I do the same thing except:

>>

>> 1) I use png, rather than jpeg, files

>>

>> 2) I do a

>> device, scale_factor=2.0

>> for PS output so that the resultant ps->png conversion doesn't

>> produce teeny pictures.

>>

>> Grabbing IDL output from screen (direct graphics at least) leaves you hostage to blechy fonts.

>

> Paul,

>

> Would you be willing to provide a small code snippet to show

> us exactly how this is done? I'd be happy to publish it for

> you. :-)

I assume you're pulling my whizzer here since I know I don't do anything that isn't already known. Maybe it's just a convention thing on my part?

At any rate, for the record I do the following sort of thing in my .pro file:

```
pro plot_this
; Some plotting setup stuff
font   = (!d.name eq 'PS') ? 1  : -1
thick   = (!d.name eq 'PS') ? 2  : 1
charsize = (!d.name eq 'PS') ? 1.5 : 1.0
; My data
n = 1000L
x = dindgen(n)/double(n-1)
y = x^2
; Do the plots
plot, x, y, $
      font=font, thick=thick, charsize=charsize
end
```

When I want to look at/play with the data onscreen, I simply do

IDL> plot_this

To create "regular" PS output I just do

```
IDL> pson
IDL> plot_this
IDL> psoff
```

and the resultant idl.ps file contains a plot that I can e.g. insert in my latex documents. When I want to create a png for use on a web page I just do,

```
IDL> pson
IDL> device, scale_factor=2.0
IDL> plot_this, x, y
IDL> psoff
```

and use the ImageMagick utility like so

```
IDL> $convert idl.ps idl.png
```

and the idl.png is large enough to read stuff.

I think the big trick here may simply be religious use of Liam Gumley's pson and psoff utilities - they take care of most things PS related. Or maybe it's the usual slew of

```
font    = (!d.name eq 'PS') ? 1  : -1
thick    = (!d.name eq 'PS') ? 2  : 1
charsize = (!d.name eq 'PS') ? 1.5 : 1.0
```

that I do. That idiom I use often enough I'm thinking of putting it in an include file, let's call it "plotset_1x1.pro" for a single plot. I could then use the same settings for other single plots:

```
pro plot_that
  @plotset_1x1
  n = 1000L
  x = dindgen(n)/double(n-1)
  y = x^4 - 10.0d0*x^3
  ; Do the plots
  plot, x, y, $
    font=font, thick=thick, charsize=charsize
end
```

I just ran a test of all the above and it worked just fine. Probably a "plotset" that was !p.multi aware and returned a structure that could be passed in via the _extra to the plot command would be more general and handier, but the above works for me 95% of the time (trying to satisfy the remaining 5% would probably lead to something like iTools :o)

All this would be much simpler if IDL didn't change the @%\$^#% charsize when multiple plots are displayed. Whoever thought that little beauty up (or, at the very least, decided that the user didn't need control over it) should've been tossed out on his/her earhole

during the pre-implementation code review.... :o)

Anyway.....

cheers,

paulv
