Subject: Re: Easy way to make hard copies at full printer resolution
Posted by Kristian Kjaer on Wed, 02 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Sounds great! Any chance of getting hold of this code, preferably with
the ten minutes added in?

- Kristian

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by davidf on Wed, 02 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Kristian Kjaer (kristian.kjaer@risoe.dk) writes:

> One thing that makes IDL somewhat inconvenient for _interactive_ data
> display
> and reduction is the lack of a command (equivalent to - dare I say it -
> alt(File)...Print... in Windows programs) to produce a hard copy of the
> current plot with the resolution offered by the printer.
>
> Ideally I'd like to sit and type into the IDL command prompt, using
> native IDL
> commands and my own (wrapper, mainly) routines and, when a useful plot
> resulted on the screen, make a hard copy of it without any extra
> trouble.
>
> Does anyone have some ideas or some useful code?

At the Object-Oriented IDL Programming course I taught in Albuquerque
a couple of weeks ago we wrote, as an exercise, a forward and
backward connected linked list object. We then used that list
object in a "smart" image object so that the image could
"remember" (in any arbitrary order) what processing steps
had been executed on it. Among the numerous benefits of this
smart image was that fact that we had a multiple "undo"
feature, since we could simply delete the last item on
its "memory" list and tell the processed image to "recreate"
itself. Very, very nice.

I've since done some more work on the linked list object
and added the capability to sort, order, and interactively
edit the linked list objects. What I have been envisioning
is exactly the functionality you seem to be asking for:
a way to store, edit, and re-play a series of commands.

What is missing (and what could be added in about 10

minutes of work) is a command "player". This would simply
be an object that executed and stored the commands for
later playback. The PRINT method would look something
like this:

```
PRO Player::Print
  Set_Plot, 'PRINTER'
  Device, XSize=5, YSize=4, /Inches
  self.history->Replay
  Device, /Close_Document
END
```

The commands to be executed would have to be entered like
this:

```
player->cmd, 'Plot, findgen(11), Title=First Plot'
player->cmd, 'y = Sin(findgen(11))+ 0.5'
player->cmd, 'Oplot, y'
```

The CMD method would look something like this:

```
PRO Player::CMD, command
  WSet, self.wid
  ok = Execute(command)
  IF NOT ok THEN RETURN
  self.history->Add, command
END
```

There would, of course, be a few more complications than
this (you might, for example, have to have a way to let
the player know that you erased the plot), but nothing
that couldn't be sorted out in about half a day's work,
I imagine.

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Easy way to make hard copies at full printer resolution

Posted by Kristian Kjaer on Thu, 03 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

(The shareware site sounds like a good thing.)

The command grabber/replayer should be very useful.
However, I think there are some frequently-occurring situations when this would not be the method of choice for generating a hard copy, e.g.,
> * plot, <very complicated, time-consuming expression> ,or
> * plot,y & y=some_function(y) & oplot,y ; if you replay this, then y ; will be further modified.

Plotting object graphics instead of direct graphics would do it, but then, as I understand it, all the nice IDL commands (plot, contour, surface, xyouts,..., with their numerous optional keywords) are not available but would have to be programmed in terms of atoms.

The really effective, neat solution would be if IDL would make it so that all the usual commands can be used to plot into an object graphic or (almost the same thing, I suppose)
if, when plotting direct graphics to the screen, one could optionally have the resulting graphic vector primitives stored in a buffer for subsequent rendering on a hard-copy device.

Maybe, I we shout loud enough here, IDL will see the light and implement it in the next version?

- Kristian