Subject: Re: PostScript and IDL,
Posted by davidf on Tue, 26 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Lars (larsh@magnet.drcmr.dk) writes:

> Hi, I'm having problems with PostScript generated by IDL_5.1 on a
> Linux system -- I expect the problem to be general, though.
>
> In principle, I'd like to use the Live_tools and other procedures
> (like Insight) that allows easy printing of plots already on screen.
>
> The generated PostScript is, however, rather useless since even the
> simplest plot generates very large PostScript files. The problem seems
> to be that the screen buffer is simply copied, rather than redrawing
> the plot to the printer device. A very large bitmap is therefore
> written instead of a few PostScript primitives.

This is correct.

> I wonder, if there is a way to redraw a view generated by a procedure
> like live_plot to a direct graphics device? Other suggestions? Any
> help is appreciated.

There is absolutely no way to draw object graphics in a
direct graphics window and visa versa. The two systems
are completely separate and distinct. Currently, there
is also no way to write object graphics out to a PostScript
file in the same format that is used, for example, with
direct graphics commands.

There is hope, however. Dick and I have been experimenting
with writing LiveTool-like programs that use object programming
techniques and direct graphic commands. As I have mentioned
here previously, we have been amazed at what is possible.
The huge advantage of writing these programs as objects
is that they are so incredibly easy to extend. Want another
feature? Just add a simple method.

A "plot object" we have recently written for a client
has amazing capability. So much so that if I hadn't
written it myself I would have a hard time believing that
IDL was behind it. Among its many advantages is that it is
very fast and it can draw itself in a PostScript file.

You can also take a bit of care in the way you write
direct graphics programs so that you can automatically
send them to a PostScript file. Most of the second half

of my IDL Programming Techniques book explains what these principles are. Many of the programs available on my web page also illustrate the technique. (See, for example, the XWINDOW program, with which it is fairly easy to build your own "live tools" applications). I am also currently devoting a large portion of my IDL Programming Techniques courses (offered through RSI this year) to learning the advantages of object programming with direct graphics.

Dick and I have been kicking around the idea of writing a class library of direct graphics objects that we could offer for sale. But unfortunately, it appears to be a shrinking market. As computers (and, presumably, printers) get faster and faster the object graphics advantages begin to dominate. It is not at all clear to us which marketplace offers the most commercial advantages in the medium term.

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/