
Subject: 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](#)

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.

There seem to be only the possibilities of

- * dumping the screen (resolution limited by screen) or
- * `set_plot,'ps'` & ; repeat all the previous plot-commands

In the (now sadly deceased) DOS-program GENPLOT printer-resolution hardcopy

was offered very simply and effectively by the program storing all the vector

strokes since the last screen erase. Then, something like

`GENPLOT> hc dev postscript`

would produce a *.ps file and optionally spool it.

Now, why doesn't IDL offer something similar, I wonder.

I thought about journaling everything and parsing the journal file to extract

a `redraw.pro` so that

`IDL> @redraw.pro`

would redraw the last plot on the currently selected device but this would

work poorly in case of, e.g.,

- * drawing to more than one window, or
- * `plot, <very complicated, time-consuming expression>` ,or
- * `plot,y & y=some_function(y) & oplot,y`

Then there is IDL5's 'Object Graphics'. I leafed through the manual and, from

what little I could understand, this rather takes the `_interactive_` out of

IDL.

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 has

resulted on the screen, make a hard copy of it without any extra trouble.

Does anyone have some ideas or some useful code?

- Thanks, Kristian

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

Posted by [mirko_vukovic](#) on Thu, 03 Sep 1998 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article <35EE44D0.75836E2F@risoe.dk>,

Kristian Kjaer <kristian.kjaer@risoe.dk> wrote:

> (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, if we shout loud enough here, IDL will see the light and implement

> it in the next version?

>

> - Kristian

>

The way I approached the problem (the solution is broken now, as my version of LinkedList (object) is back in the shop to make sure the cleanup is done correctly) has a plotw object that stores all the data that it is supposed to plot. It can then re-plot them at will. In the meantime, it will accept changes in coordinate system (right now it deals with polar and cartesian 2D only), line types and so on.

In an ideal world I would have posted it, but never found the time to package it (it relies on a slew of home-grown routines) and write a user guide.

Ideally, if I give it out, I would like any modifications to it to be done in a self-consistent way (something like emacs).

If you want it I can try to email it, but be prepared to sweat it out for a couple of days untill it works as advertised :-)

Now on to credits (not all of this is my idea).

The first form of this idea came to me when I had to transpose the axis of a very complicated plot. I found it easier to send then the plot commands to an object that would then do the axis-setting and plotting.

A couple of months later, D.Fanning posted/emailed his latest/greatest idea about IDL direct graphic plots as objects. This started my thinking.

I needed something that could accept polar/rectangular data and plot in polar/rectangular coordinates. Being impatient, I put the PlotW object in about a week, 10 days. (It also deals with simultaneous multiple windows, plot regions, and such, remembering the sysvars, pixmaps, if requested). One day I may even implement Smith Charts with it.

From David's post it looks that we did not quite follow the same paths, which is great. I did not breathe too easy writing the code based on his ideas.

Mirko

-----== Posted via Deja News, The Leader in Internet Discussion ==-----
http://www.dejanews.com/rg_mkgrp.xp Create Your Own Free Member Forum

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [davidf](#) on Thu, 03 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

I wrote just a couple of minutes ago:

- > Live tools are the right idea,
- > but for reasons that I don't even fully understand I really
- > don't like to use them. Perhaps because I can't see the
- > code to understand and modify how they work. But when
- > I work with them I find myself becoming increasingly
- > annoyed rather than increasingly satisfied. "My God, things
- > just don't lay out in the window properly!" Perhaps I
- > should see a code shrink. :-)

Having finished this rant, I grabbed the nearest magazine and headed off for my morning meditation. I found myself reading this: "Feng shui reveals how our environment

can nurture or cripple our contentment, personal growth, health, and relationships."

What those damn Live Tools lack is Feng shui!

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: Re: Easy way to make hard copies at full printer resolution
Posted by [davidf](#) on Thu, 03 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

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

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

The next version!? I think not. But we can always hope for the version after that. I personally don't see it happening at all, although I do hold out hope that the direct and object graphics might become more integrated, especially with respect to PostScript output.

If object printing problems are resolved, I see everyone migrating to object graphics functions over time. The only real drawback is that we need more high-level plotting tools so we don't have to write two pages of code every time we want to do something simple. Live tools are the right idea, but for reasons that I don't even fully understand I really don't like to use them. Perhaps because I can't see the code to understand and modify how they work. But when I work with them I find myself becoming increasingly annoyed rather than increasingly satisfied. "My God, things just don't lay out in the window properly!" Perhaps I should see a code shrink. :-)

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: Re: Easy way to make hard copies at full printer resolution
Posted by [davidf](#) on Fri, 06 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Liam Gumley (Liam.Gumley@ssec.wisc.edu) writes:

> It's going to take some work, but the person(s) that
> develop this product will go down in history as IDL
> heroes.

That's what I'm saying. Here is the perfect opportunity
for some young buck (or doe, for that matter. How come
there are so few women on this newsgroup? Thank goodness
for Rose!) to make a huge name for him or her self.

Goodness knows, we couldn't possibly offer you any more
clues. :-)

Cheers,

David

P.S. So your marriage goes to hell because of all
those hours you will be spending in the evening.
Think of the GLORY!

--

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

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [Liam Gumley](#) on Fri, 06 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Kristian Kjørlund" wrote:
[a very reasonable wish list]

I don't see any reason why IDL could not offer the kind of functionality you describe (Insight doesn't cut it). It's going to take some work, but the person(s) that develop this product will go down in history as IDL heroes.

Cheers,
Liam.
PS: Too bad I've already got a full time job.

--

Liam E. Gumley
Space Science and Engineering Center, UW-Madison
<http://cimss.ssec.wisc.edu/~gumley>

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [davidf](#) on Fri, 06 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

> Does anyone know about any other user-routines along these lines?

I've outlined several times in this group how such a program could be written as an object, and I've even tested and confirmed my theory in a program for a client, but no one has taken the bait and taken on the job of writing it up for the newsgroup. Surely someone out there has a strong desire to be rich and famous.

Oh, well...right, famous, anyway. You aren't going to get rich giving software away. :-)

> And, why is it that IDL cannot get round to offering this kind of
> functionality built-in?

Too easy. They know if they provide this kind of functionality IDL programmers would get totally bored and start learning C++ just to have something to do. *Then* where would they be?

Cheers,

David

P.S. Let's just say that if everyone who wanted a program

like this was willing to pay me \$15 for it, I'd write the darn thing myself over the weekend. :-)

--

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

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [davidf](#) on Fri, 06 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Craig Markwardt (craigmnet@cow.physics.wisc.edu) writes:

- > I would *love* it if the graphics window would simply remember what's
- > been plotted, and be able to dump that out to the postscript device.
- > A little print button at the top of the window would do wonders. Does
- > this work in Windows?

Absolutely! Why do you think we all switched from UNIX?
Oh, whoops, hang on. I just tested this. Bummer. :-(

- > All of the programs found by Kristian are written in IDL, so
- > necessarily try to solve the problem at the *front* end, e.g. by
- > intercepting PLOT commands. It would be much more simple and elegant
- > to solve the problem at the back end in the graphics driver. Has
- > anybody ever written a graphics driver?

Uh, no, I haven't written a graphics driver. But I am tired of hearing about it and I could use someone pounding me on the back and telling me what a great guy I am (I lost another tennis match today), so I'm gonna have a go at writing something. Probably won't be this weekend, however. The middle son and I are sitting through 18 excruciating hours of soccer referee instruction this weekend. No wonder those idiots can't see a foul. They are brain-dead by the end of the first weekend!

Anyway, I've got a few ideas... :-)

Cheers,

David

--

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

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [Craig Markwardt](#) on Fri, 06 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

>
> My apologies if this is turning into a FAQ:
>
> 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 MS-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.

I would *love* it if the graphics window would simply remember what's
been plotted, and be able to dump that out to the postscript device.
A little print button at the top of the window would do wonders. Does
this work in Windows?

Sure, it says somewhere in the IDL documentation that this would take
too much memory. Maybe five years ago, but we aren't in the stone
ages any more.

All of the programs found by Kristian are written in IDL, so
necessarily try to solve the problem at the *front* end, e.g. by
intercepting PLOT commands. It would be much more simple and elegant
to solve the problem at the back end in the graphics driver. Has
anybody ever written a graphics driver?

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [Kristian Kjaer](#) on Fri, 06 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

- > P.S. Let's just say that if everyone who wanted a program
- > like this was willing to pay me \$15 for it, I'd write the
- > darn thing myself over the weekend. :-)

What? My problem solved? For a mere USD 15 ? You're on!

- Kristian

PS: I *do* realize this is only my IDL problem you'll solve!

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [Kristian Kjaer](#) on Sat, 07 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Stein Vidar Hagfors Haugan wrote:

- > Probably the best thing to start with is the JOURNAL
- > command. The system variable !JOURNAL is set to the

I made the enclosed jnl.pro, and I also put it in my startup-file.

The purpose is to save a few keystrokes, make sure you're allways
journaling, and that you can easily close the journal file and
start a new one.

(I seem to recall that in IDL on (NT-)windows, FLUSH doesn't seem to
work.)

But I didn't get to the point where I could machine-edit the
journal-file.

- Kristian

```
; Begin code -----  
pro jnl, close=close, stop=stop, $  
    automatic=automatic, new=new, flush=flush, $  
    query=query, help=help, usage=usage, $  
    debug=debug, replace=replace, $  
    filename=filename, text=text, atext  
;PURPOSE:    jnl.pro is a wrapper routine for JOURNAL  
;CALLING SEQUENCE:  
;    jnl, [atext,] /<keywords>  
;INPUTS:  
;    atext      : Filename or text to journal (optional)  
;KEYWORD PARAMETERS:  
;MODIFICATION HISTORY:  
;    K. Kjaer    Riso Nov-98:
```

;-

```
true=1
sep='/'
if (!version.os_family eq 'Windows') then sep='\

case true of
(keyword_set(close) or keyword_set(stop)) and (!JOURNAL gt 0) : begin
; Close the journal file:
fstat_journal=fstat(!JOURNAL)
journal_file=findfile(fstat_journal.name) ; truncated if IDLversion lt
5.
JOURNAL
Print, 'Closed journal file '+fstat_journal.name+' ('+journal_file+')'
end
(keyword_set(close) or keyword_set(stop)) and (!JOURNAL eq 0) : begin
Print, 'Not journalling'
end
keyword_set(query) and (!JOURNAL eq 0) : begin
Print, 'Not journalling'
end
keyword_set(query) and (!JOURNAL gt 0) : begin
if keyword_set(verbose) then help,/str,fstat(!JOURNAL)
fstat_journal=fstat(!JOURNAL)
journal_file=findfile(fstat_journal.name) ; truncated if IDLversion lt
5.
Print, 'Journalling to '+fstat_journal.name+' ('+journal_file+')'
end
keyword_set(flush) and (!JOURNAL gt 0) : begin
flush,!journal ; Seems to have no effect!
end
keyword_set(help) or keyword_set(usage) : begin
Print,'Usage: jnl, /close, /stop, /automatic, /new, /flush, /query,
/help, /usage,/debug, /replace, filename="...", text="...",
"filename_without_blanks", "text with blanks in it"
end
(keyword_set(new) or keyword_set(automatic)) and (!JOURNAL gt 0) : begin
jnl, debug=debug, replace=replace,/close
jnl, debug=debug, replace=replace,/new
end
(keyword_set(new) or keyword_set(automatic)) and (!JOURNAL eq 0) : begin
journal_file=getenv('temp') ; the temp directory
if keyword_set(debug) then print, 'temp directory: '+journal_file
if (journal_file eq "") then begin
print, 'temp is not assigned'
print, 'Pls. assign an environment variable temp to point to
your chosen scratch directory!'
```

```

    journal_file='.'
endif
if (strmid(journal_file,strlen(journal_file)-1,1) ne sep) then
journal_file=journal_file+sep
if keyword_set(debug) then print, journal_file
; Generate a new, unique file name (I'm sorry that, so far this is
systime() converted to 8.3 format):
journal_file=journal_file+string(systime(1)*.1,format='(f12. 3)')
; Another approach could be based on bin_date(systime(0))
if keyword_set(debug) then print, journal_file
jnl, debug=debug, replace=replace,filename=journal_file
end
exist(filename) and (!JOURNAL gt 0) : begin
jnl, debug=debug, replace=replace,/close
jnl, debug=debug, replace=replace,filename=filename
end
exist(filename) and (!JOURNAL eq 0) : begin
; Open the journal file:
; (1st check if filename already present !)
findfilename=findfile(filename,count=count)
if ((count gt 0) and not keyword_set(replace)) then begin
Print, 'Journal file '+filename+' already present. Use
jnl,...,/replace or another file name !'
Print, 'Not journalling.'
endif else begin
JOURNAL,filename
fstat_journal=fstat(!JOURNAL)
journal_file=findfile(fstat_journal.name) ; truncated if IDLversion
lt 5.
Print, 'Started journalling to '+fstat_journal.name+'
('+journal_file+)'
endelse
end
exist(text) and (!JOURNAL gt 0) : begin
JOURNAL,TEXT
end
exist(text) and (!JOURNAL eq 0) : begin
jnl, debug=debug, replace=replace,/query
end
exist(atext) : begin
if (strcompress(strtrim(atext,2),/remove_all) eq atext) then $
jnl, debug=debug, replace=replace,filename=atext else $
jnl, debug=debug, replace=replace,text =atext
end
else: begin
jnl, debug=debug, replace=replace,/help
jnl, debug=debug, replace=replace,/query
end

```

endcase
end
; End code -----

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [Kristian Kjaer](#) on Sat, 07 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Craig Markwardt wrote:

> A little print button at the top of the window would do wonders. Does
> this work in Windows?

Sure, for excel, origin, CoreDraw, you-name-it, ...
(They call it WYSIWYG).
But not for IDL_for_windows.

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [Kristian Kjaer](#) on Sat, 07 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Stein Vidar Hagfors Haugan wrote:

> Probably the best thing to start with is the JOURNAL

.....

> fancy_journal_editing,'printwinexec.pro'
> @printwinexec.pro

That a nice, simple idea.

The trouble might be,

a) if

IDL> y=<some_complicated_and_time_consuming_expression> & plot,x

then y would be evaluated twice. And, worse,

b) if y was created before the beginning of time, and

IDL> plot,y & y=y+1 & oplot,y

then the hard copy would be different from the screen plot.

- Kristian

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [davidf](#) on Sat, 07 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Stein Vidar Hagfors Haugan (steinhh@ulrik.uio.no) writes:

> I've got a few myself, if you do the work, David!

> Probably the best thing to start with is the JOURNAL
> command.

An interesting idea, Stein Vidar, but the thought of editing journal commands leaves me cold for some reason. Perhaps there is ample room here for two approaches to the same problem. :-)

I am thinking along the lines of a more powerful XWindow program, but object based. And as a matter of fact, I have the basic idea working (I.e., a program that takes and displays an unlimited number of commands) in just a little over an hour of programming last night when I got back from referee training. (This is why I LOVE objects!) Now it is just a matter of fleshing it out and adding the error checking and documentation. Should be ready by the first of the year. :-)

> Oops, I better go back to my thesis before I get
> carried away like I did with the dlmform script.

Ah, I wondered what had become of you lately. I've missed you. Well, good writing! :-)

> So, when will we see it, David :-)

Soon, I believe. Real soon. Get your checkbooks ready. :-)

Cheers,

David

--

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

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [steinhh](#) on Sat, 07 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <MPG.12150a3fd1c555a989880@news.frii.com>
davidf@dfanning.com (David Fanning) writes:

> Uh, no, I haven't written a graphics driver. But I am
> tired of hearing about it and I could use someone pounding me
> on the back and telling me what a great guy I am (I lost
> another tennis match today), so I'm gonna have a go at
> writing something. [...]

Well, that would be great, indeed!

> Anyway, I've got a few ideas... :-)

I've got a few myself, if you do the work, David!
Probably the best thing to start with is the JOURNAL
command. The system variable !JOURNAL is set to the
file unit used for journalling, you could use that to
FLUSH,!JOURNAL to make sure everything's there. Grab
the file contents, let the user set the starting point
(or possibly auto-detect e.g. wset/window and similar
procedures!) and edit a little bit, then squeeze the
program in between your favourite PS "wrappers" and
execute.. probably at the \$main\$ level!

So, users should invoke your script with e.g. "@printwin",
and @printwin may contain e.g.:

```
fancy_journal_editing,'printwinexec.pro'  
@printwinexec.pro
```

You could of course use RECALL_COMMANDS instead - though
the default of 20 lines stored is not enough. The user
has to either start JOURNAL, or change !EDIT_INPUT
to have sufficient storage.... Oops, I better go back
to my thesis before I get carried away like I did
with the dlmform script..

So, when will we see it, David :-)

Cheers,

Stein Vidar

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [Craig Markwardt](#) on Mon, 09 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Craig Markwardt <craigmnet@cow.physics.wisc.edu> writes:
> It would be much more simple and elegant
> to solve the problem at the back end in the graphics driver. Has

> anybody ever written a graphics driver?

In the money-where-your-mouth-is department, I've had a go at writing a device driver that captures graphics commands in IDL. It uses a trick which seems to work: I install a new "X" device driver which replaces the original one, does some specialized processing, and then passes the command on to the original true X driver. You might consider it to be a "parasitic" driver since it doesn't do any real X output.

At the moment, all it does is intercept the commands, print them to the console, and then pass them down the chain. Despite this, it's still quite interesting to see how graphics calls work on IDL. See "How to activate" below to load it into your IDL session via `call_external`.

You are welcome to try it out and provide suggestions. While the code looks lengthy, that's just because I trap all graphics primitives. It works under Linux IDL version 5.X, though I expect it should work under any Unix version with the right compilation line. (See `$IDL/external/call_external/C/callex_unix.txt` for help). An intrepid Mac/Windows programmer might get it to work too.

No other practical issues, like storing commands, redirecting them to postscript, etc are addressed yet. Before I forge ahead I thought I'd see if anybody else had ideas.

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

----- Beginning xfilter.c
/*

XFILTER - device driver which captures IDL output (DRAFT)

Copyright (C) 1999 Craig B. Markwardt

This C code implements a graphics device "filter" for IDL. The intent is to capture graphics commands which can be reproduced at a later time. For example, to capture the graphics that appear on the X windows display for printing later.

Currently this works only under X windows and Unix (Linux is tested).

At present the graphics commands are simply echoed to the IDL standard error stream, not stored.

The method is simple. A "new" graphics device is registered. IDL's notion of a graphics device is a jump table which implements various drawing primitives. Each primitive is implemented below, but trivially calls the true X device primitive. Thus the devices are chained together. Both graphics and windowing primitives are chained.

See below for compilation instructions (for Linux) and how to activate this within IDL.

IDL is a trademark of RSI.

```
*/

#include <stdio.h>
#include "export.h"

/*
  Compile with:
  gcc -g -fPIC xfilter.c -shared -o xfilter.so
*/
/*
  Activate in IDL with:
  a = call_external(getenv('PWD')+'/xfilter.so', 'xfilter', 0)
  set_plot, 'x'
*/

#define oprintf(x...) fprintf(stderr, x);

extern IDL_DEVICE_DEF _IDL_x_dev; /* IDL X windows device driver */
IDL_DEVICE_DEF  xfilter_dev; /* New device driver block */
IDL_DEVICE_CORE  *xdev, *xfdev;
IDL_DEVICE_WINDOW *xwin, *xfwin;

int cattrq = 0, ctextq = 0; /* Flags determine when attributes are cached */
IDL_ATTR_STRUCT cattr; /* Cached graphics attributes */
IDL_TEXT_STRUCT ctext; /* Cached text attributes */

/*****
/*
    Graphics primitives
*/

/* Draw line segment */
void xfilter_draw(IDL_GR_PT *p0, IDL_GR_PT *p1, IDL_ATTR_STRUCT *a)
{
    int reset = 0;
```

```

IDL_GR_PT xp0, xp1, *pp0 = 0, *pp1 = 0;

/* Determine if we need to recache graphics attributes */
if (!cattrq) reset = 1;
else if ((a->color != cattr.color) || (a->thick != cattr.thick)
|| (a->linestyle != cattr.linestyle) || (a->chl != cattr.chl))
    reset = 1;

/* Recache */
if (reset) {
    oprintf("ATTR: COLOR=%d, THICK=%d, LINESTYLE=%d, CHANNEL=%d\n",
        a->color, a->thick, a->linestyle, a->chl);
    cattr = *a;
    cattrq = 1;
}

oprintf("DRAW:");
if (p0) {
    oprintf(" (X0=%d,Y0=%d)", p0->i.x, p0->i.y);
}
if (p1) {
    oprintf(" (X1=%d,Y1=%d)", p1->i.x, p1->i.y);
}
oprintf("\n");

xdev->draw(p0, p1, a); /* Call original X driver */
}

/* Draw hardware text */
int xfilter_text(IDL_GR_PT *p0, IDL_ATTR_STRUCT *a,
    IDL_TEXT_STRUCT *ta, char *t)
{
    int reset = 0;
    IDL_GR_PT xp0, *pp0 = 0;

    /* Determine if we need to recache graphics attributes */
    if (!cattrq) reset = 1;
    else if ((a->color != cattr.color) || (a->thick != cattr.thick)
|| (a->linestyle != cattr.linestyle) || (a->chl != cattr.chl))
        reset = 1;

    /* Reset cache */
    if (reset) {
        oprintf("ATTR: COLOR=%d, THICK=%d, LINESTYLE=%d, CHANNEL=%d\n",
            a->color, a->thick, a->linestyle, a->chl);
        cattr = *a;
        cattrq = 1;
    }
}

```

```

/* Determine if we need to recache text attributes */
reset = 0;
if (!ctextq) reset = 1;
else if ((ta->font != ctext.font) || (ta->axes != ctext.axes)
|| (ta->size != ctext.size) || (ta->orien != ctext.orien)
|| (ta->align != ctext.align))
    reset = 1;

/* Recache */
if (reset) {
    oprintf("TEXT_ATTR: FONT=%d, AXES=%d, SIZE=%f, ORIEN=%f, ALIGN=%f\n",
        ta->font, ta->axes, ta->size, ta->orien, ta->align);
    ctext = *ta;
    ctextq = 1;
}

oprintf("TEXT: ");
if (p0) {
    oprintf(" (X0=%d,Y0=%d)", p0->i.x, p0->i.y);
}
oprintf(" '%s'\n", t);
return xdev->text(p0, a, ta, t); /* Call original X driver */
}

/* Erase graphics viewport */
void xfilter_erase(IDL_ATTR_STRUCT *a)
{
    int reset = 0;

    /* Determine if we need to recache graphics attributes */
    if (!cattrq) reset = 1;
    else if ((a->color != cattr.color) || (a->thick != cattr.thick)
|| (a->linestyle != cattr.thick) || (a->chl != cattr.chl))
        reset = 1;

    /* Recache */
    if (reset) {
        oprintf("ATTR: COLOR=%d, THICK=%d, LINESTYLE=%d, CHANNEL=%d\n",
            a->color, a->thick, a->linestyle, a->chl);
        cattr = *a;
        cattrq = 1;
    }
    oprintf("ERASE\n");
    xdev->erase(a); /* Call original X driver */
}

/* Capture cursor information */

```

```

void xfilter_cursor(int funct, IDL_MOUSE_STRUCT *m)
{
    oprintf("CURSOR\n");
    xdev->cursor(funct, m); /* Call original X driver */
}

/* Filled polygons. Yuck */
void xfilter_polyfill(int *x, int *y, int n, IDL_POLYFILL_ATTR *poly)
{
    oprintf("POLYFILL\n");
    xdev->polyfill(x, y, n, poly); /* Call original X driver */
}

/* Called when returning to interactive level; graphics state should be
   well defined after this call.
   */
void xfilter_inter_exit(void)
{
    oprintf("INTER_EXIT\n");
    catrq = 0; ctextq = 0;
    xdev->inter_exit(); /* Call original X driver */
}

/* Flush any pending graphics output to the display */
void xfilter_flush(void)
{
    oprintf("FLUSH\n");
    catrq = 0; ctextq = 0;
    xdev->flush(); /* Call original X driver */
}

/* Load new color map? Doesn't seem to be used */
void xfilter_load_color(IDL_LONG start, IDL_LONG n)
{
    oprintf("LOAD_COLOR:%d %d\n", start, n);
    xdev->load_color(start, n); /* Call original X driver */
}

/* Read or write a pixmap to the display */
void xfilter_rw_pixels(UCHAR *data, int x0, int y0, int nx,
                      int ny, int dir, IDL_TV_STRUCT *secondary)
{
    oprintf("RW_PIXELS:%d %d %d %d %d\n", x0, y0, nx, ny, dir);
    /* Call original X driver */
    xdev->rw_pixels(data, x0, y0, nx, ny, dir, secondary);
}

/* Implement DEVICE command */

```

```

void xfilter_dev_specific(int argc, IDL_VPTR *argv, char *argk)
{
    int i;
    oprintf("DEVICE: BEGIN\n");
    for(i=0; i<argc; i++) if (argv[i]->type) IDL_Print(1, &argv[i], 0);
    oprintf("DEVICE: END\n");

    xdev->dev_specific(argc, argv, argk); /* Call original X driver */
}

/* Implement HELP, /DEVICE */
void xfilter_dev_help(int argc, IDL_VPTR *argv)
{
    xdev->dev_help(argc, argv); /* Call original X driver */
}

/* Called once, when device driver is installed */
void xfilter_load_rtn(void)
{
    oprintf("LOAD_RTN\n");
    /* Don't call the original driver, since it expects to be called once (?) */
    /* xdev->load_rtn(); */
}

/*****
/*          Window  primitives          */

/* Create Window */
void xfilter_window_create(int argc, IDL_VPTR *argv, char *argk)
{
    oprintf("WINDOW_CREATE -> ");
    xwin->window_create(argc, argv, argk);
    oprintf("%d\n", _IDL_x_dev.window);

    ctextq = 0; cattrq = 0; /* Uncache */
}

void print_int(IDL_VPTR p)
{
    switch(p->type) {
    case IDL_TYP_BYTE:    oprintf("%dB \n", (int) p->value.c); break;
    case IDL_TYP_INT :    oprintf("%d \n", (int) p->value.i); break;
    case IDL_TYP_LONG:    oprintf("%dL \n", (int) p->value.l); break;
    case IDL_TYP_UINT:    oprintf("%dU \n", (int) p->value.ui); break;
    case IDL_TYP_ULONG:   oprintf("%dUL\n", (int) p->value.ul); break;
    default:              oprintf("default\n"); break;
    }
}

```

```

/* Delete Window */
void xfilter_window_delete(int argc, IDL_VPTR *argv)
{
    oprintf("WINDOW_DELETE: ");
    print_int(argv[0]);

    ctextq = 0; cattrq = 0; /* Uncache */
    xwin->window_delete(argc, argv);
}

/* Show (activate) window */
void xfilter_window_show(int argc, IDL_VPTR *argv, char *argk)
{
    oprintf("WINDOW_SHOW: ");
    print_int(argv[0]);

    xwin->window_show(argc, argv, argk);
}

/* Set new active window */
void xfilter_window_set(int argc, IDL_VPTR *argv)
{
    oprintf("WINDOW_SET: ");
    print_int(argv[0]);

    ctextq = 0; cattrq = 0; /* Uncache */
    xwin->window_set(argc, argv);
}

/* Window menuing (?) */
IDL_VPTR xfilter_window_menu(int argc, IDL_VPTR *argv, char *argk)
{
    oprintf("WINDOW_MENU\n");
    return xwin->window_menu(argc, argv, argk);
}

/* Initialization routine */
long int xfilter(int argc, char *argv[])
{
    IDL_SYSFUN_DEF *f;

    /* Copy device driver */
    xfilter_dev = _IDL_x_dev;
    ctextq = 0; cattrq = 0;

    /* Convenience pointers, makes things easier to type above */
    xdev = &xfilter_dev.core;

```

```

xfdev = &_IDL_x_dev.core;
xwin  = &xfilter_dev.winsys;
xfwin = &_IDL_x_dev.winsys;

/* Set up new jump table - graphics primitives */
xfdev->draw      = &xfilter_draw;
xfdev->text      = &xfilter_text;
xfdev->erase     = &xfilter_erase;
xfdev->cursor    = &xfilter_cursor;
xfdev->polyfill  = &xfilter_polyfill;
xfdev->inter_exit = &xfilter_inter_exit;
xfdev->flush     = &xfilter_flush;
xfdev->rw_pixels = &xfilter_rw_pixels;
xfdev->dev_specific = &xfilter_dev_specific;
xfdev->dev_help  = &xfilter_dev_help;
xfdev->load_rtn  = &xfilter_load_rtn;

/* Set up new jump table - window primitives */
xfwin->window_create = &xfilter_window_create;
xfwin->window_delete = &xfilter_window_delete;
xfwin->window_show   = &xfilter_window_show;
xfwin->window_set    = &xfilter_window_set;
xfwin->window_menu   = &xfilter_window_menu;

/* Add device */
IDL_AddDevice(&_IDL_x_dev, 1);

return 0;
}

```

Subject: Re: Easy way to make hard copies at full printer resolution
 Posted by [Nick Bower](#) on Mon, 09 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

```

> I would *love* it if the graphics window would simply remember what's
> been plotted, and be able to dump that out to the postscript device.
> A little print button at the top of the window would do wonders. Does
> this work in Windows?
>
> All of the programs found by Kristian are written in IDL, so
> necessarily try to solve the problem at the *front* end, e.g. by
> intercepting PLOT commands. It would be much more simple and elegant
> to solve the problem at the back end in the graphics driver. Has
> anybody ever written a graphics driver?

```

I do agree that it would be nice to have RSI address the problem so it doesn't have to be addressed by users at the front-end. But I also feel

it's more fundamental than improving one or two device back-ends, and it would be nice to have it addressed by RSI as a direct graphics issue. imho, the direct graphics as a whole is a little painful in this area, and the plotting procedures for any device are ridiculously linear in issuing plotting instructions. Everyone knows that if you simply want to change the axis scale, you have to go back and issue all the plot and oplot commands all over again. Not sure how other people feel, but I was personally amazed by this when learning IDL.

So it seems the best way to achieve this functionality is to write your own objects (again, a front-end approach as you pointed out), but that's just not a good time-investment for most users. So perhaps there is a market a good commercial front-end package as David suggested. (I haven't seen any open source initiatives in my limited observation of this group so am ruling out that). I wouldn't mind seeing a fairly decent first cut at it however rather than something specific to the postscript device only.

- > Sure, it says somewhere in the IDL documentation that this would take
- > too much memory. Maybe five years ago, but we aren't in the stone
- > ages any more.

I also agree the memory argument just doesn't cut the mustard anymore. RSI are being slack here.

nick

--

Nick Bower
Space Science and Engineering Center
University of Wisconsin - Madison
<http://arm1.ssec.wisc.edu/~nickb>

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [Mirko Vukovic](#) on Mon, 09 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have an animal that does the trick (call it plotw), but it is not fully tamed. It accepts overplots, it can change scales, delete plot components, do simple geometric transformations, plot in rect/polar, replot, print, send to ps, return cursor positions, accept parent widgets, etc.

In addition, this is a routine that organically grew, and is in need of a re-write. And there are definitely bugs. I started

sketching a new one at home, but
it is going slowly.

However I do not have a place where to post it.
Thus, to make it public, I would need the following assistance:
Somewhere where to place it, and also, a beta tester. This person
would have to do a couple of trial runs to make sure that I did ship all
the routines that
plotw needs. For example, it uses a linked list object among others.

So it would take a few tries until the package is self-sufficient.

Any helpers?

Mirko

Sent via Deja.com <http://www.deja.com/>
Share what you know. Learn what you don't.

Subject: Re: Easy way to make hard copies at full printer resolution
Posted by [Theo Brauers](#) on Mon, 09 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Kristian Kjørlund" wrote:

>
> Craig Markwardt wrote:
>> A little print button at the top of the window would do wonders. Does
>> this work in Windows?
>
> Sure, for excel, origin, CoreDraw, you-name-it, ...
> (They call it WYSIWYG).
> But not for IDL_for_windows.

The print button exists not in IDL but in ghostview. I create postscript
files all the time and have them updated in the ghostview window instead
of
re-executing the code with and without device set to PS. Works fine for
me.

I would appreciate if the direct graphics window would have a buffer
for postscript which can be send to the printer. This buffer could hold
more
than one frame and could be send to the printer/ghostview on request.

Theo

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

Posted by [Nick Bower](#) on Fri, 13 Aug 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

> Kristian> Ideally I'd like to sit and type into the IDL command
> Kristian> prompt, using native IDL commands and my own (wrapper,
> Kristian> mainly) routines and, when a useful plot resulted on
> Kristian> the screen, make a hard copy of it without any extra
> Kristian> trouble.
>
> Why wouldn't setting the device to 'ps', doing your plot to a temp
> file, then shelling a 'gs' or 'gv' command work? Your button could
> just lpr the temp file.
>
> Or, if you have to have the window in an IDL canvas, plot to 'ps',
> execute 'gs' with your screen resolution and output to a TIFF, then
> read the temp file and TV to your display.

Interesting. But I think Kristian had a valid point there. You'll
always be able to find find work arounds, but perhaps it's also sensible
to question the functions provided in original product, considering the
price tag.

want to print? do everything twice or else write your own software
wrappers - it doesn't make sense to me.

nick

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

Posted by [David Fenyes](#) on Fri, 13 Aug 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Kristian> Ideally I'd like to sit and type into the IDL command
Kristian> prompt, using native IDL commands and my own (wrapper,
Kristian> mainly) routines and, when a useful plot resulted on the
Kristian> screen, make a hard copy of it without any extra
Kristian> trouble.

Why wouldn't setting the device to 'ps', doing your plot to a temp
file, then shelling a 'gs' or 'gv' command work? Your button could
just lpr the temp file.

Or, if you have to have the window in an IDL canvas, plot to 'ps',
execute 'gs' with your screen resolution and output to a TIFF, then
read the temp file and TV to your display.

Alternatively, why can't you write plot wrappers that plot to the screen, and save the command to a temp string, then 'print screen' could set device to 'ps', execute the temp string, and print the temp file.

Probably one of these could be done with simple wrappers and good display quality both in IDL and your printer.

Regards,

David.

--

David Fenyes -- dfenyes@home.com

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

Posted by [davidf](#) on Mon, 16 Aug 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Martin Schultz (m218003@modell3-d.dkrz.de) is putting his money where his mouth is when he comments:

```
>> Soon, I believe. Real soon. Get your checkbooks ready. :-)  
>  
> you are guaranteed $15 from me if I may comment on your programs  
> afterwards ;-)
```

Martin, your comments are worth **more** than money. You might get your version free. :-)

Cheers,

David

P.S. My office is in shambles this morning from the installation of some new windows. But I'm working on it.... :-)

--

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: davidf@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

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

Posted by [m218003](#) on Mon, 16 Aug 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

> Soon, I believe. Real soon. Get your checkbooks ready. :-)

>

> Cheers,

>

> David

>

David,

you are guaranteed \$15 from me if I may comment on your programs afterwards ;-)

Cheers,

Martin.

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

Posted by [Mirko Vukovic](#) on Mon, 16 Aug 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article <m3yafgd5ze.fsf@c34680-a.grlnd1.tx.home.com>,

David Fenyes <dfenyes@flash.net> wrote:

>

> Kristian> Ideally I'd like to sit and type into the IDL command

> Kristian> prompt, using native IDL commands and my own (wrapper,

> Kristian> mainly) routines and, when a useful plot resulted on the

> Kristian> screen, make a hard copy of it without any extra

> Kristian> trouble.

>

> Why wouldn't setting the device to 'ps', doing your plot to a temp

> file, then shelling a 'gs' or 'gv' command work? Your button could

> just lpr the temp file.

>

> Or, if you have to have the window in an IDL canvas, plot to 'ps',

> execute 'gs' with your screen resolution and output to a TIFF, then

> read the temp file and TV to your display.

>

> Alternatively, why can't you write plot wrappers that plot to the

> screen, and save the command to a temp string, then 'print screen'

> could set device to 'ps', execute the temp string, and print the temp

> file.

>

> Probably one of these could be done with simple wrappers and good

> display quality both in IDL and your printer.

>

> Regards,
>
> David.
> --
> David Fenyes -- dfenyes@home.com
>

I have a routine that does that, but was built for in-house use and is thus fairly rough (see my previous post).

Since then Prof. David Klassen has offered space on his web page for this routine. So I am currently cleaning it up (actually re-writing, since it is so much fun) and hope to have a beta version ready in two weeks. (It will be command driven, not yet widgeted).

It will most probably not produce the type of plots that you may want. But I am trying to write it in a way that makes it easily extendable -- meaning no need to get into the routine and add code (that was the problem with the old version, it was growing like the Thing), but write objects that follow certain definitions, and then call the routine with keywords, again that follow certain definitions.

So, stay tuned. I work on this at 5:30 in the mornings (mostly) for an hour or so, so it is taking time.

Mirko

Sent via Deja.com <http://www.deja.com/>
Share what you know. Learn what you don't.
