Subject: Windows IDL DLM with Cygwin
Posted by joeymu on Mon, 11 Jul 2005 23:02:09 GMT

Has anyone sucessfully used a DLM created with the Cygwin family of
compilers? I have sucessfully compiled a DLM and it looks like it
should work, but when it gets to the point of actually using the DLM,
IDL will crash! I've been trying to compile my code with a
-mno-cygwin, but that is going to be a lot of work! I'm hoping there
is an easier way. Anyway, I'd love to hear even if this is possible
(i.e. someone has actually done it) so I don't just spin my wheels.

Thanks!

Joey

Subject: Re: Windows IDL DLM with Cygwin
Posted by joey on Wed, 13 Jul 2005 14:24:55 GMT

Haje Korth <haje.korth@jhuapl.edu> wrote:

> To Joey: I would not waste my precious research time to try to get cygwin
> compile your code. This is a painful lession I learned during my times as a
> graduate student. Even though it is from M$, I had a lot of luck with
> Visual Studio. Most code especially research code is generic enough not
> to rely OS specific extensions, so that I never had to spend much time
> on conversions.

Our code is a bit of a hybrid of lots of legacy things. It is a very large
body of code for accessing some data files and automatically retrieving said
data from a remote database. This is why it has all the unixism's in it.
In fact, cygwin can actually compile all of our code and everything works
with Windows. Its just interfacing with IDL that is causing me problems.

Getting Visual Studio to compile everything will be very difficult, but
not impossible so I may do it after all.

>> shows that it *is* possible to use Cygwin-compiled DLLs with IDL, via
>> CALL_EXTERNAL. I don't see any mention of the -mno-cygwin switch in the
>> techtip, so it looks like these are true Cygwin DLLs.
>>
>> Of course DLMs are another matter, but this techtip suggests that maybe
>> you should persist.

Is there a good reference on the difference between DLM and CALL_EXTERNAL?
Someone told us to make a DLM and we did so I never researched the possibility

of doing a CALL_EXTERNAL. Possibly I could convert my code from DLM to a
CALL_EXTERNAL to keep the Windows people happy?

Does anyone know if GDL supports either of these mechanisms for using
external code?

Thanks,
Joey

Subject: Re: Windows IDL DLM with Cygwin
Posted by Haje Korth on Wed, 13 Jul 2005 14:42:54 GMT
View Forum Message <> Reply to Message

GDL ist not supported under Windows, so getting that to compile will be even
harder. Reading you description on what you are trying to do, you are not
talking UDF DLM, are you?

Haje

<joey@swri.edu> wrote in message news:11da91ni4i7qf93@corp.supernews.com...
> Haje Korth <haje.korth@jhuapl.edu> wrote:
>
>> To Joey: I would not waste my precious research time to try to get cygwin
>> compile your code. This is a painful lession I learned during my times as
>> a
>> graduate student. Even though it is from M$, I had a lot of luck with
>> Visual Studio. Most code especially research code is generic enough not
>> to rely OS specific extensions, so that I never had to spend much time
>> on conversions.
>
> Our code is a bit of a hybrid of lots of legacy things. It is a very
> large
> body of code for accessing some data files and automatically retrieving
> said
> data from a remote database. This is why it has all the unixism's in it.
> In fact, cygwin can actually compile all of our code and everything works
> with Windows. Its just interfacing with IDL that is causing me problems.
>
> Getting Visual Studio to compile everything will be very difficult, but
> not impossible so I may do it after all.
>
>>> shows that it *is* possible to use Cygwin-compiled DLLs with IDL, via
>>> CALL_EXTERNAL. I don't see any mention of the -mno-cygwin switch in the
>>> techtip, so it looks like these are true Cygwin DLLs.
>>>
>>> Of course DLMs are another matter, but this techtip suggests that maybe

```
>>> you should persist.
>
> Is there a good reference on the difference between DLM and CALL_EXTERNAL?
> Someone told us to make a DLM and we did so I never researched the
> possibility
> of doing a CALL_EXTERNAL.  Possibly I could convert my code from DLM to a
> CALL_EXTERNAL to keep the Windows people happy?
>
> Does anyone know if GDL supports either of these mechanisms for using
> external code?
>
> Thanks,
> Joey
```

Subject: Re: Windows IDL DLM with Cygwin
Posted by Mark Hadfield on Wed, 13 Jul 2005 22:04:17 GMT

View Forum Message <> Reply to Message

joey@swri.edu wrote:
> Is there a good reference on the difference between DLM and CALL_EXTERNAL?
> Someone told us to make a DLM and we did so I never researched the possibility
> of doing a CALL_EXTERNAL. Possibly I could convert my code from DLM to a
> CALL_EXTERNAL to keep the Windows people happy?

Read RSI's "External Development Guide". You should be able to find it
somewhere in the help system.

I have personal experience only with DLLs that are to be called via
CALL_EXTERNAL. This technique is described in the EDG under "Techniques
that don't use IDL's Internal API". It is reasonably easy to construct
such DLLs. The pitfall is that the information passed to and from the
DLL is not checked so:

  - You will need to write an IDL wrapper program that knows the
    data types to be passed and does the checking.

  - If you get it wrong, IDL crashes.

It takes more work to write a DLM (Dynamically Loadable Modules) but its
integration into IDL is much better.

--
Mark Hadfield   "Kei puwaha te tai nei, Hoea tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: window
Posted by Ken Mankoff on Wed, 02 Nov 2005 23:45:51 GMT

On Wed, 2 Nov 2005, ChiChiRuiz@gmail.com wrote:
> Hi, I was wondering how to change the default when calling
> "window". I'd like to have a white background with colored plots,
> rather than the default black background with white-color plots.

You can set !p.background to the index in the CT you want. Not sure
about foreground. However, I leave those alone and customize the CT
itself like this:

```
TVLCT, r,g,b, /GET
r[0] = ( g[0] = ( b[0] = 255 ) )
r[255] = ( g[255] = ( b[255] = 0 ) )
TVLCT, r,g,b
```

> Besides, without calling "loadct", I can specify color=250 for a
> red color plot, how do i get a yellow/blue/green color by simply
> using color=(a number)? Thanks for your help.

I cannot replicate this. For me the default CT is grayscale with
index 0 as black and 255 as white so color=250 is almost-white.
Perhaps you are running a site-wide startup file that is
initializing a CT for you with red at index 250?

If you want yellow/blue/green color you need to
  a) make sure they are in your colortable
  b) figure out where they are
  c) then use the color=(a number) syntax to access them.

  -k.

Subject: Re: window
Posted by Kenneth P. Bowman on Thu, 03 Nov 2005 02:37:28 GMT

In article <Pine.OSX.4.61.0511021840310.6909@gouda.local>,
 Ken Mankoff <mankoff@gmail.com> wrote:

> On Wed, 2 Nov 2005, ChiChiRuiz@gmail.com wrote:
>> Hi, I was wondering how to change the default when calling
>> "window". I'd like to have a white background with colored plots,
>> rather than the default black background with white-color plots.
>
> You can set !p.background to the index in the CT you want. Not sure

> about foreground.

To change the foreground color set !P.COLOR.

I wish it were possible to do this in my startup.pro and have it stick,
but if you change devices (to PS, for example) and then change back, it
reverts to the default white-on-black. (This could be solved by using
routines like Liam Gumley's PSON and PSOFF to handle the device changes.)

IDL Version 6.1, Mac OS X (darwin ppc m32). (c) 2004, Research Systems,
Inc.

```
IDL> print, !p.color, !p.background
   16777215        0
IDL> !p.color = 0
IDL> print, !p.color, !p.background
        0        0
IDL> set_plot, 'ps'
IDL> print, !p.color, !p.background
        0      255
IDL> set_plot, 'x'
IDL> print, !p.color, !p.background
   16777215        0
```

Since I generally want screen plots to look as much as possible like
Postscript output, I just set !P.COLOR and !P.BACKGROUND whenever I set
up a display window.

Is now a good time to make my perennial complaint that the PS driver
really needs to fully support 24-bit color?

Ken Bowman

Subject: Re: window
Posted by David Fanning on Thu, 03 Nov 2005 06:47:03 GMT
View Forum Message <> Reply to Message

ChiChiRuiz writes:

> Hi, I was wondering how to change the default when calling "window".
> I'd like to have a white background with colored plots, rather than the
> default black background with white-color plots. Besides, without
> calling "loadct", I can specify color=250 for a red color plot, how do
> i get a yellow/blue/green color by simply using color=(a number)?

!P.Background = FSC_Color('ivory')
!P.Color = FSC_Color('navy')

Plot, indgen(11), /NoData
Oplot, indgen(11), Color=FSC_Color('blue')
Oplot, Reverse(indgen(11)), Color=FSC_Color('green')

You can find FSC_COLOR here:

  http://www.dfanning.com/programs/fsc_color.pro

You are getting a red plot, because you are in color
decomposition turned on mode, which is the default mode
for IDL running with a 24-bit graphics card. You could
set yourself into INDEXED color mode by typing this:

  Device, Decomposed=0

But if you use FSC_Color to specify your colors there
is no need to. You will get the specified color no matter
what color decomposition state you choose to work in.
(And why would anyone NOT want to display all 16.7 million
colors simultaneously!? Don't listen to the Luddites who
want you to set yourself in PSUEDO_COLOR mode!)

You might want to browse the numerous articles on using
color in IDL on my web page. In particular, you want to
read this one:

  http://www.dfanning.com/color_tips/colors24.html

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: window
Posted by David Fanning on Thu, 03 Nov 2005 06:48:37 GMT
View Forum Message <> Reply to Message

Ken Mankoff writes:

> You can set !p.background to the index in the CT you want. Not sure
> about foreground. However, I leave those alone and customize the CT
> itself like this:
>
> TVLCT, r,g,b, /GET

> r[0] = (g[0] = (b[0] = 255))
> r[255] = (g[255] = (b[255] = 0))
> TVLCT, r,g,b

Yikes! Do you ever find youself producing white on white
PostScript files when you do this? :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Ken Mankoff writes:

> I cannot replicate this. For me the default CT is grayscale with
> index 0 as black and 255 as white so color=250 is almost-white.
> Perhaps you are running a site-wide startup file that is
> initializing a CT for you with red at index 250?

Try this, you will see what he sees:

    Device, Decomposed=1
    Plot, indgen(11), Color=250

> If you want yellow/blue/green color you need to
>   a) make sure they are in your colortable
>   b) figure out where they are
>   c) then use the color=(a number) syntax to access them.

Good advice for the 1970's, but the last time I looked
it was at least 2001. :-)

There are MUCH better ways to specify colors these days.
But why do we always have to roll our own!? Couldn't RSI
figure out a way to give us some simple drawing colors?

Cheers,

David

P.S. Sorry if I appear to be a little out of sorts.
It happens to me every time I teach an IDL class and
have to apologise all day long about windows that don't
resize, and colors that are incomprehensible, TV commands
that don't display images in color, etc, etc. Shouldn't
someone be paying attention to the essentials that people
use every day instead of dreaming up yet more iTool functionality
that no one I know wants to use? Just my 2 cents worth of ranting.
--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: window
Posted by ChiChiRuiz@gmail.com on Thu, 03 Nov 2005 14:55:26 GMT
View Forum Message <> Reply to Message

Thanks David. By the way, it's a pleasure to read your book! (I just
started reading last week).

Subject: Re: window
Posted by ChiChiRuiz@gmail.com on Thu, 03 Nov 2005 15:57:32 GMT
View Forum Message <> Reply to Message

> P.S. Sorry if I appear to be a little out of sorts.
> It happens to me every time I teach an IDL class and
> have to apologise all day long about windows that don't
> resize, and colors that are incomprehensible, TV commands
> that don't display images in color, etc, etc. Shouldn't
> someone be paying attention to the essentials that people
> use every day instead of dreaming up yet more iTool functionality
> that no one I know wants to use? Just my 2 cents worth of ranting.

Hahahah, that's so funny. I totally agree with you. I used to use
Matlab, and plotting with colors and symbols and resizing and zooming
are soooo much easier. I just got those emails about iTool...so I know
what you mean...

Angie

Subject: Re: window
Posted by Mark Hadfield on Thu, 03 Nov 2005 19:08:33 GMT

David Fanning wrote:
> There are MUCH better ways to specify colors these days.
> But why do we always have to roll our own!? Couldn't RSI
> figure out a way to give us some simple drawing colors?

No need. You've done it for them. How much did they pay you, by the way?

--
Mark Hadfield   "Kei puwaha te tai nei, Hoea tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: window
Posted by David Fanning on Thu, 03 Nov 2005 23:04:35 GMT

Mark Hadfield writes:

> No need. You've done it for them. How much did they pay you, by the way?

I don't know. I'm still hauling the money to the bank. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/