
Subject: Re: IDLgrLegend geometry
Posted by [davidf](#) on Mon, 26 Feb 2001 23:46:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

George Constantinides (gconstantinides@mhl.nsw.gov.au) writes:

> I was resizing an IDLgrWindow object and noticed that objects such as
> IDLgrAxis, IDLgrPlot, IDLgrText resized correctly to fit the new
> dimensions, but IDLgrLegend did not.
> This topic has been discussed in the past but with the demise of the
> DejaNews news group database, I could not find any references.
> So here is a work around that works well on the screen (IDLgrWindow) but
> when you send the View to another output object such as IDLgrPrinter,
> the fixed geometry of IDLgrLegend becomes a problem again.

Humm. It's pretty clear that the author of IDLgrLegend was *trying* to solve this problem in the ComputeDimensions method. Have you tried using this when you resize or change to another destination device? It looks to me like this would solve all your problems, although I confess I've never used IDLgrLegend.

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: IDLgrLegend geometry
Posted by [Mark Hadfield](#) on Tue, 27 Feb 2001 00:17:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.150498b583111c38989d69@news.frii.com...
> George Constantinides (gconstantinides@mhl.nsw.gov.au) writes:
>
>> I was resizing an IDLgrWindow object and noticed that objects such as
>> IDLgrAxis, IDLgrPlot, IDLgrText resized correctly to fit the new
>> dimensions, but IDLgrLegend did not.
>> [...]
>
> Humm. It's pretty clear that the author of IDLgrLegend
> was *trying* to solve this problem in the ComputeDimensions

> method. Have you tried using this when you resize or change
> to another destination device? It looks to me like this
> would solve all your problems, although I confess I've
> never used IDLgrLegend.

From my reading of the IDLgrLegend code in version 5.4, ComputeDimensions is called every time the legend is re-drawn, so there should be no need to call it manually. ComputeDimensions recalculates the dimensions of the legend's atoms based on the character size of the legend text, and since the text objects are created with RECOMPUTE_DIMENSIONS = 2, the legend should be resized every time it is drawn.

In fact on my system legends do get resized as the size of the destination device changes! I wonder why they don't on yours, George. What version are you using? Do you have a line that looks like this in the CreateGlyphs method in idlgrlegend__define.pro?

```
(*self.pTexts)[index] = OBJ_NEW('IDLgrText', $  
                                FONT = self.oFont, $  
                                COLOR = (*self.pText_Color), $  
                                STRINGS = (*self.pltem_Name)[index],$  
                                RECOMPUTE_DIMENSIONS = 2)
```

If you want to debug this you could put a break in ComputeDimensions and see what's happening. (Warning: you may be surprised how often a model's Draw method is called.)

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield>
National Institute for Water and Atmospheric Research

Subject: Re: IDLgrLegend geometry
Posted by [George Constantinides](#) on Tue, 27 Feb 2001 04:26:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Hadfield wrote:

> "David Fanning" <davidf@dfanning.com> wrote in message
> news:MPG.150498b583111c38989d69@news.frii.com...
>> George Constantinides (gconstantinides@mhl.nsw.gov.au) writes:
>>
>>> I was resizing an IDLgrWindow object and noticed that objects such as
>>> IDLgrAxis, IDLgrPlot, IDLgrText resized correctly to fit the new
>>> dimensions, but IDLgrLegend did not.
>>> [...]
>>

>> Humm. It's pretty clear that the author of IDLgrLegend
>> was *trying* to solve this problem in the ComputeDimensions
>> method. Have you tried using this when you resize or change
>> to another destination device? It looks to me like this
>> would solve all your problems, although I confess I've
>> never used IDLgrLegend.

I tried calling ComputeDimensions explicitly but I was not making any sense
of the result so I gave up on this approach.

>
>
> From my reading of the IDLgrLegend code in version 5.4, ComputeDimensions is
> called every time the legend is re-drawn, so there should be no need to call
> it manually. ComputeDimensions recalculates the dimensions of the legend's
> atoms based on the character size of the legend text, and since the text
> objects are created with RECOMPUTE_DIMENSIONS = 2, the legend should be
> resized every time it is drawn.

>
> In fact on my system legends do get resized as the size of the destination
> device changes! I wonder why they don't on yours, George. What version are
> you using? Do you have a line that looks like this in the CreateGlyphs
> method in idlgrlegend__define.pro?
>
> (*self.pTexts)[index] = OBJ_NEW('IDLgrText', \$
> FONT = self.oFont, \$
> COLOR = (*self.pText_Color), \$
> STRINGS = (*self.pltem_Name)[index],\$
> RECOMPUTE_DIMENSIONS = 2)
>
> If you want to debug this you could put a break in ComputeDimensions and see
> what's happening. (Warning: you may be surprised how often a model's Draw
> method is called.)
>

I am using 5.4 on NT.

Now that I know it works for someone I'll go and have another look.

George Constantinides
Manly Hydraulics Laboratory
URL <http://www.mhl.nsw.gov.au>

Subject: Re: IDLgrLegend geometry
Posted by [davidf](#) on Tue, 27 Feb 2001 05:33:28 GMT

Mark Hadfield (m.hadfield@niwa.cri.nz) writes:

```
> In fact on my system legends do get resized as the size of the destination
> device changes! I wonder why they don't on yours, George. What version are
> you using? Do you have a line that looks like this in the CreateGlyphs
> method in idlgrlegend__define.pro?
>
>     (*self.pTexts)[index] = OBJ_NEW('IDLgrText', $
>                                     FONT = self.oFont, $
>                                     COLOR = (*self.pText_Color), $
>                                     STRINGS = (*self.pltem_Name)[index],$
>                                     RECOMPUTE_DIMENSIONS = 2)
```

Speaking of the RECOMPUTE_DIMENSIONS keyword, I realized in the last object class I taught that my understanding of what this keyword does was exactly the opposite of what it *actually* does. (I discovered this when some overzealous student actually typed the commands I said to type and discovered that the program did the opposite of what I said it would do. I *hate* students like this.)

This whole question suddenly rang some bells with me, and I spent some time this evening going through the documentation very s-l-o-w-l-y, trying to understand it.

Text characters are sized according to a text "box", whose width and height are given in the "data" units of your arbitrary coordinate system. (See the CHAR_DIMENSIONS keyword.) What RECOMPUTE_DIMENSIONS can do is tell you when to recompute the size of that text box. For example, if you change the data range, you will probably want to recompute your text box.

But, and here is the point I was confused about, if you are just re-sizing the graphic there is no need to recompute the text box, since the data range doesn't change at all. In fact, in resizing windows you explicitly do NOT want to recompute dimensions.

You can see this by downloading the Simple_Surface program from my web page:

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

In this program, I have RECOMPUTE_DIMENSIONS set to 2. Notice when you resize the window that the text

sizes remain the same size. (I still doesn't understand why this should be so, and I am looking for enlightenment on this point.) But if you change all the RECOMPUTE_DIMENSIONS=2 to RECOMPUTE_DIMENSIONS=0 you will find that the text is size proportionally to the axes, the data, etc. This is the behaviour I want.

So, (sorry for the stream of consciousness here, it is late and I am very tired), why, if I don't change the data coordinate system at all, simply resize the window and then recompute the text box dimensions, does the text **always** stay the same size? Is it because a 14 point font is a particular size regardless of the size of the output window?

Mark!? Are you back from lunch yet? :-)

Cheers,

David

P.S. I think IDLgrLegend works correctly because the author is calculating new text box sizes for each draw. Thus, he **should** recompute the dimensions before every draw. But this seems VERY low-level to me. Is this really necessary?

--

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: IDLgrLegend geometry

Posted by [Mark Hadfield](#) on Tue, 27 Feb 2001 20:29:31 GMT

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

"David Fanning" <davidf@dfanning.com> wrote in message news:MPG.1504ea1077595651989d6a@news.frii.com...

> In this program, I have RECOMPUTE_DIMENSIONS set to
> 2. Notice when you resize the window that the text
> sizes remain the same size. (I still doesn't understand
> why this should be so, and I am looking for enlightenment
> on this point.) But if you change all the RECOMPUTE_DIMENSIONS=2
> to RECOMPUTE_DIMENSIONS=0 you will find that the text
> is size proportionally to the axes, the data, etc. This

> is the behaviour I want.

Oops. I am guilty (again) of not thinking and observing carefully enough before I post. RECOMPUTE_DIMENSIONS does work exactly as you say and I think that is the opposite of what I implied in my earlier message. However, unlike you, David, I prefer my text to stay the same size when I resize my windows so I tend to use RECOMPUTE_DIMENSIONS = 2.

BTW the legend test program I referred to earlier is mgh_example_legend, of which there is a copy at

<http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/>

but don't bothering downloading it right now because somebody told me it's broken. I will fix it shortly.

Anyway, when I run mgh_example_legend (fixed version) and resize the window, all the symbols (including the ones in the legend) scale with the window and all the text (including the legend labels) keeps a constant size. This behaviour seems reasonable to me.

Looking back at your original post, George, I am not sure what you mean by "resize correctly to fit the new dimensions". Do you want the legend text to scale with the window size? Perhaps you should try changing the IDLgrLegend code so that text objects are created with RECOMPUTE_DIMENSIONS = 0.

This business of being an IDL guru is very hard, what with making yourself look silly all the time and having people tell you your code is broken. I think I'll take a breather. How do you manage it, David?

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield>
National Institute for Water and Atmospheric Research

Subject: Re: IDLgrLegend geometry
Posted by [Pavel A. Romashkin](#) on Thu, 01 Mar 2001 20:01:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Hadfield wrote:

> This business of being an IDL guru is very hard, what with making yourself
> look silly all the time and having people tell you your code is broken. I
> think I'll take a breather. How do you manage it, David?

You can always jump to the lurkers category :-)

