
Subject: IDLgrLegend geometry

Posted by [George Constantinides](#) on Mon, 26 Feb 2001 22:53:19 GMT

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

Hi all,

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.

```
;-----
```

```
PRO Resize_Events, event
```

```
    ; resize event handler.
```

```
Widget_Control, event.top, Get_UValue=info
```

```
    ; Get old window dimension
```

```
info.thisWindow->GetProperty, Dimension=WDimOld
```

```
    ; Resize the draw widget.
```

```
info.thisWindow->SetProperty, Dimension=[event.x, event.y]
```

```
    ; Resize the Legend. Calculate the size based on the x-width of the window
```

```
    ; This is a workaround to get the legend font to resize.
```

```
    ; It works well on the screen (IDLgrWindow) but not on the printer (IDLgrPrinter).
```

```
info.oLegend->GetProperty, Font=IFont
```

```
Font->GetProperty, Size=IChar
```

```
IFont->SetProperty, Size=IChar*event.x/WDimOld[0]
```

```
    ; Redisplay the graphic.
```

```
Widget_Control, Hourglass=1
```

```
info.thisWindow->Draw, info.oScene  
Widget_Control, Hourglass=0
```

```
;Put the info structure back.
```

```
Widget_Control, event.top, Set_UValue=info, /No_Copy
```

```
END
```

```
;-----
```

Does anyone else have the same problem ?

If not, then what is the "proper" way to resize legends ?

If yes, then I am glad that I am not the only one. Did you solve the problem irrespective of the output destination object, or you gave up in discussed ?

George Constantinides
Manly Hydraulics Laboratory
email: GeorgeC@mhl.nsw.gov.au
URL http://www.mhl.nsw.gov.au

Subject: Re: IDLgrLegend geometry
Posted by [davidf](#) on Tue, 27 Feb 2001 21:02:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

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

> 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?

I've had a lot of practice, being married to a woman who
is very good at puncturing overinflated egos. :-(

Cheers,

David

P.S. I have to tell you, though, I am relishing
the thought of Carol coming home tonight. She has
been named the Distinguished Teacher of the Year at

her school. And we are all hard at work writing glowing letters of recommendation to be included in a big scrapbook that will be presented to Carol at a big-deal dinner in a couple of weeks.

I took mine over to the school where Carol has taught for 12+ years today and handed it to the lady in the front office. "Hi, can I leave this for Carol Seemueller, please," I asked.

"Who is Carol Seemueller?", she replied.

I can't wait to tell Carol that story. It will make up for 100's of stories like that she tells about me. :-)

--

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 [George Constantinides](#) on Wed, 28 Feb 2001 04:32:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
> 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)
>
>> Now that I know it works for someone I'll go and have another look.
>
> It's puzzling that it works for me & not for you. It may be something to do
> with the fact that my test program has several layers of non-standard code
> between the IDLgrLegend and the IDLgrWindow.
```

I changed RECOMPUTE_DIMENSIONS=0 and recompiled IDLgrLegend__Define.pro and it resized the way I expected (ie. the sizes of the legend text increased

proportionately to the new window size.) I will not use this as a permanent fix though but it does advance my understanding of IDLgrLegend.

It appears that the default legend behavior is not consistent with other objects such as IDLgrAxis, IDLgrText. I would have expected that, if you are resizing a Window, all Text based objects either stay the same size OR all Text objects resize proportionately to the new output object. As it is, some resize and others don't even though I have not explicitly set the RECOMPUTE_DIMENSIONS for any of them.

```
> 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.
```

The IDL documentation is not clear on this issue but the use of RECOMPUTE_DIMENSIONS in "simple_surface.pro" allows me to have another go at the interpretation:

```
<online help for IDLgrText>
;+
;RECOMPUTE_DIMENSIONS (Get, Set)
;
```

;Set this keyword to one of the following values to indicate when this text
;object's character dimensions (refer to the CHAR_DIMENSIONS property)
;are to be recomputed automatically:
;

; the CHAR_DIMENSIONS property. If CHAR_DIMENSIONS is set to [0,0],
; compute once and re-use the resulting dimensions until the
; CHARACTER_DIMENSIONS are modified.

;-

When =0, Dave covers this pretty well in his post.

When =2, IDL recomputes the dimensions to maintain the size of the text when
applying the new transformation matrix. This is the size that was originally
set when the object was created or modified by the
oText->SetProperty Size=xxx method.

I hope, I am not stating the obvious here. Feel free to correct me if I have
completely missed the point.

--

Regards,
George Constantinides
Manly Hydraulics Laboratory
email: GeorgeC@mhl.nsw.gov.au
URL <http://www.mhl.nsw.gov.au>

Subject: Re: IDLgrLegend geometry
Posted by [davidf](#) on Wed, 28 Feb 2001 05:43:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

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

> It appears that the default legend behavior is not consistent with other
> objects such as IDLgrAxis, IDLgrText. I would have expected that, if you
> are resizing a Window, all Text based objects either stay the same
> size OR all Text objects resize proportionately to the new output object.
> As it is, some resize and others don't even though I have not explicitly set
> the RECOMPUTE_DIMENSIONS for any of them.

Well, as the discussion with Mark and me points out,
some like it hot, and some like it cold. What the
author of the program probably wishes he had done
is allowed the user to set a keyword that would give
the user whatever he or she wants.

> I hope, I am not stating the obvious here. Feel free to correct me if I have

> completely missed the point.

I don't think you have missed the point. Although I'm pretty sure I have been missing the point for quite a while now. I **think** I understand it now. I'll find out for sure the next time I try to teach it to someone. :-(

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
