
Subject: FG Bug -- Legend/Cleanup

Posted by [Matthew Argall](#) on Wed, 04 Mar 2015 21:02:41 GMT

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

So, here are a couple bugs I found a few days ago. I try to make a legend with the RELATIVE keyword, but it is rejected. I then destroy the window object, and the widget is destroyed except for the title bar and the buttons bar. When I click the "X" to close the remaining parts of the window, I get the PSTATE error. I have to reset the IDL session to get rid of the window pieces.

This happens any time legend does not like a keyword (i.e. if something is mistyped).

```
IDL> win = window()
IDL> p1 = plot(/TEST, /CURRENT)
IDL> leg1 = legend(TARGET=p1, POSITION=[1.0, 1.0], HORIZONTAL_ALIGNMENT='RIGHT',
VERTICAL_ALIGNMENT='TOP', /RELATIVE)
% LEGEND: Unknown property: RELATIVE
% Execution halted at: $MAIN$
IDL> obj_destroy, win
```

[Attempt to close remaining part of the window by clicking "X"]

```
% Pointer type required in this context: PSTATE.
% Execution halted at: $MAIN$
```

```
IDL> help, !version
** Structure !VERSION, 8 tags, length=104, data length=100:
  ARCH      STRING  'x86_64'
  OS        STRING  'darwin'
  OS_FAMILY  STRING  'unix'
  OS_NAME    STRING  'Mac OS X'
  RELEASE    STRING  '8.2'
  BUILD_DATE STRING  'Apr 10 2012'
  MEMORY_BITS INT     64
  FILE_OFFSET_BITS
                INT     64
```

Subject: Re: FG Bug -- Legend/Cleanup

Posted by [Matthew Argall](#) on Wed, 04 Mar 2015 21:30:19 GMT

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

Another bug:

In legend, I can specify an array of target objects, each with different text colors (actually, the text color keyword changes the sample line color, not the text color)

```
win = window()
p1 = plot(/TEST, /CURRENT, LAYOUT=[1,2,1], NAME='Plot 1', COLOR='Black')
p2 = plot(/TEST, /CURRENT, LAYOUT=[1,2,2], NAME='Plot 2', COLOR='Blue')
leg1 = legend(TARGET=[p1, p2], LABEL='Plot X', TEXT_COLOR=['Black', 'Blue'])
```

But if I try to specify two labels, I get an error

```
win = window()
p1 = plot(/TEST, /CURRENT, LAYOUT=[1,2,1], NAME='Plot 1', COLOR='Black')
p2 = plot(/TEST, /CURRENT, LAYOUT=[1,2,2], NAME='Plot 2', COLOR='Blue')
leg1 = legend(TARGET=[p1, p2], LABEL=['Plot X', 'Plot Y'], TEXT_COLOR=['Black', 'Blue'])
% Expression must be a scalar in this context: <BYTE    Array[2]>.
```

I either have to change the label after the legend is created or add the label separately

```
leg1[1].LABEL = 'Plot Y'
```

```
leg1 -> Add, p2, LABEL='Plot Y', TEXT_COLOR='Blue'
```

Subject: Re: FG Bug -- Legend/Cleanup

Posted by [Matthew Argall](#) on Wed, 04 Mar 2015 22:02:24 GMT

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

Another bug:

I was working with sample data that resulted in a single-point average with standard deviation. I tried to create an error plot with a legend and got an error

```
win = window()
p1 = plot(/TEST, /CURRENT, LAYOUT=[1,2,1], NAME='Plot')

y = [5.0]
yerror = [0.329]
x = [1.0]
p2 = errorplot(x, y, yerror, LAYOUT=[1,2,2], NAME='ErrorPlot 2', /CURRENT,
SYMBOL='Diamond', X RANGE=[0,2], Y RANGE=[0,10])
leg1 = legend(TARGET=p1, LABEL='Plot X', TEXT_COLOR='Black')
leg1 -> Add, p2, LABEL='Plot Y', TEXT_COLOR='Blue'
% Variable is undefined: ONEWLEGENDITEMS.
% Execution halted at: $MAIN$
```

Subject: Re: FG Bug -- Legend/Cleanup

Posted by [lecacheux.alain](#) on Thu, 05 Mar 2015 14:55:49 GMT

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

On Wednesday, March 4, 2015 at 11:02:26 PM UTC+1, Matthew Argall wrote:

> Another bug:

>

> I was working with sample data that resulted in a single-point average with standard deviation. I tried to create an error plot with a legend and got an error

>

> win = window()

> p1 = plot(/TEST, /CURRENT, LAYOUT=[1,2,1], NAME='Plot')

>

> y = [5.0]

> yerror = [0.329]

> x = [1.0]

> p2 = errorplot(x, y, yerror, LAYOUT=[1,2,2], NAME='ErrorPlot 2', /CURRENT, SYMBOL='Diamond', X RANGE=[0,2], Y RANGE=[0,10])

> leg1 = legend(TARGET=p1, LABEL='Plot X', TEXT_COLOR='Black')

> leg1 -> Add, p2, LABEL='Plot Y', TEXT_COLOR='Blue'

> % Variable is undefined: ONEWLEGENDITEMS.

> % Execution halted at: \$MAIN\$

I am not sure this is a bug.

If you call LEGEND this way:

leg1 = legend(TARGET=[p1,p2])

you can eventually force the labels by:

leg1[0].label = 'PLOT X'

leg1[1].label = 'PLOT Y'

Even better would be to initialize NAME in both plot calls accordingly.

alx.

Subject: Re: FG Bug -- Legend/Cleanup

Posted by [Matthew Argall](#) on Thu, 05 Mar 2015 15:09:20 GMT

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

> I am not sure this is a bug.

> If you call LEGEND this way:

> leg1 = legend(TARGET=[p1,p2])

> you can eventually force the labels by:

> leg1[0].label = 'PLOT X'

> leg1[1].label = 'PLOT Y'

> Even better would be to initialize NAME in both plot calls accordingly.

Yes, but having the object's name be the label is not ideal, since the only way I can retrieve plots from a window object is by their name. Let's say

```
name = '$\int_0^\infty e^{-zt} t^{-n} dt, \operatorname{Re}(z) \geq 0$'
```

this is ok for the legend. But let's say I am carrying around the window object and want to retrieve the legend. I would then have to remember the name each time:

```
theLegend = win['$\int_0^\infty e^{-zt} t^{-n} dt, \operatorname{Re}(z) \geq 0$']
```

I would much rather have a name that means something

```
name = 'theory'  
theLegend = win['theory']
```

This goes double since I do not know how to get a name of child graphics objects. Something like this would be nice:

```
child_names = win -> GetNames()
```
