

---

Subject: function graphics curiosity? bug?  
Posted by [Jonathan](#) on Tue, 12 Dec 2017 01:44:51 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Have a look at the following code:

```
x = findgen(128)
y = 1.0 + 0.1*randomn(seed,128)
y2 = 1.0 + 0.2 randomn(seed,128)
b = widget_base( xsize=480, xoffset=940, ysize=360, yoffset=0 )
w = widget_window( b, x_scroll_size=470, y_scroll_size=350 )
widget_control, b, /realize
widget_control, w, get_value=d
p = plot( x, y, current=d, xstyle=2, ystyle=2 )
p2 = plot( x, y2, color='red', linestyle="", symbol='+', current=d, /overplot )
d.uvalue = { x:x, y:y, y2:y2, b:b, p:p, p2:p2 }
end
```

This creates a widget window, w, under a base, b, and then places two overlapping plots within that window. Now, if the data changes for the second plot, I would like to do the following steps:

```
y3 = 1.0 + 0.2 randomn(seed,128)           ; new data
p2.delete                                   ; erases the p2 data in the plot
p2 = plot( x, y3, color='red', , linestyle="", symbol='+', current=d, /overplot )
d.uvalue.p2 = p2                            ; store the plot identifier in the window's uvalue
structure
```

The last line generates the following error message:  
% Attempt to store into an expression: Structure reference.  
% Execution halted at: \$MAIN\$

What has happened is that IDL forgot the type of d.uvalue.p2, so when I try to put a new (identical) p2 there, it rejects the attempt.

This turns out to be a huge hassle for me.  
My solution is a kludge, which is to create a new structure for d.uvalue and replace the whole thing, rather than just one element.

Is there a better, simpler way?

---

---

Subject: Re: function graphics curiosity? bug?  
Posted by [Markus Schmassmann](#) on Tue, 12 Dec 2017 10:37:33 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On 12/12/2017 02:44 AM, Jonathan wrote:  
> Have a look at the following code:  
>  
> x = findgen(128)

```

> y = 1.0 + 0.1*randomn(seed,128)
> y2 = 1.0 + 0.2 randomn(seed,128)
> b = widget_base( xsize=480, xoffset=940, ysize=360, yoffset=0 )
> w = widget_window( b, x_scroll_size=470, y_scroll_size=350 )
> widget_control, b, /realize
> widget_control, w, get_value=d
> p = plot( x, y, current=d, xstyle=2, ystyle=2 )
> p2 = plot( x, y2, color='red', linestyle='', symbol='+', current=d, /overplot )
> d.uvalue = { x:x, y:y, y2:y2, b:b, p:p, p2:p2 }
> end
>
> This creates a widget window, w, under a base, b, and then places
> two overlapping plots within that window. Now, if the data changes for the
> second plot, I would like to do the following steps:
>
> y3 = 1.0 + 0.2 randomn(seed,128) ; new data
> p2.delete ; erases the p2 data in the plot
> p2 = plot( x, y3, color='red', , linestyle='', symbol='+', current=d, /overplot )
> d.uvalue.p2 = p2 ; store the plot identifier in the window's uvalue structure
>
> The last line generates the following error message:
> % Attempt to store into an expression: Structure reference.
> % Execution halted at: $MAIN$
> What has happened is that IDL forgot the type of d.uvalue.p2, so
> when I try to put a new (identical) p2 there, it rejects the attempt.
>
> This turns out to be a huge hassle for me.
> My solution is a kludge, which is to create a new structure for
> d.uvalue and replace the whole thing, rather than just one element.
>
> Is there a better, simpler way?

```

```

; a simpler way:
p2.putData, y3

```

```

; ps: a '*' is missing in the definition of y2 & y3
y2 = 1.0 + 0.2 * randomn(seed,128)
y3 = 1.0 + 0.2 * randomn(seed,128)
;

```

Directly updating the values is also much faster than creating a new plot. I hope this help,

Markus

Subject: Re: function graphics curiosity? bug?

Posted by [Jonathan](#) on Tue, 12 Dec 2017 14:44:05 GMT

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

---

On Tuesday, December 12, 2017 at 6:37:37 AM UTC-4, Markus Schmassmann wrote:

> On 12/12/2017 02:44 AM, Jonathan wrote:

>> Have a look at the following code:

```
>>
>> x = findgen(128)
>> y = 1.0 + 0.1*randomn(seed,128)
>> y2 = 1.0 + 0.2 randomn(seed,128)
>> b = widget_base( xsize=480, xoffset=940, ysize=360, yoffset=0 )
>> w = widget_window( b, x_scroll_size=470, y_scroll_size=350 )
>> widget_control, b, /realize
>> widget_control, w, get_value=d
>> p = plot( x, y, current=d, xstyle=2, ystyle=2 )
>> p2 = plot( x, y2, color='red', linestyle='', symbol='+', current=d, /overplot )
>> d.uvalue = { x:x, y:y, y2:y2, b:b, p:p, p2:p2 }
>> end
```

>> This creates a widget window, w, under a base, b, and then places  
>> two overlapping plots within that window. Now, if the data changes for the  
>> second plot, I would like to do the following steps:

```
>>
>> y3 = 1.0 + 0.2 randomn(seed,128) ; new data
>> p2.delete ; erases the p2 data in the plot
>> p2 = plot( x, y3, color='red', , linestyle='', symbol='+', current=d, /overplot )
>> d.uvalue.p2 = p2 ; store the plot identifier in the window's uvalue structure
```

```
>>
>> The last line generates the following error message:
>> % Attempt to store into an expression: Structure reference.
>> % Execution halted at: $MAIN$
```

>> What has happened is that IDL forgot the type of d.uvalue.p2, so  
>> when I try to put a new (identical) p2 there, it rejects the attempt.

```
>>
>> This turns out to be a huge hassle for me.
>> My solution is a kludge, which is to create a new structure for
>> d.uvalue and replace the whole thing, rather than just one element.
```

```
>>
>> Is there a better, simpler way?
```

```
>
> ; a simpler way:
> p2.putData, y3
>
> ; ps: a '*' is missing in the definition of y2 & y3
> y2 = 1.0 + 0.2 * randomn(seed,128)
> y3 = 1.0 + 0.2 * randomn(seed,128)
> ; |
```

```
>
> Directly updating the values is also much faster than creating a new
```

> plot. I hope this help,  
>  
> Markus

That answers my question. Thank you.

I don't know why this is not made clear in the documentation, nor are there examples.

---