Subject: strange !P.multi[0] behavior??
Posted by astroboy.20000 on Thu, 07 Nov 2013 22:36:18 GMT
View Forum Message <> Reply to Message

Hello,

I created a structure, !pmap, that stores the !p, !x, !y, !z, and !map variables so that I could make some plots in one window, then open another window and make plots, and then go back to the first window to make more plots, starting where I left off. As I recall, it worked well.

However, it doesn't work now, which means either my memory is faulty or else I made some changes that I don't recall. I've included the relevant code below.

The point is that even though the program sub_pmap recalls the correct !p structure, that doesn't seem to affect the placement of plots. This is clear when you run the program.

I'm at a loss as to what might be going on here. This is a capability which I found quite useful in the past and which I find that I need again in a big way.

Thanks,

Mark

```
;************************************************************
;************************************************************
;***********      START  *********************************
;************************************************************
;************************************************************
```

```
pro start,xxx
;!pmap structure is defined


window,1,xsize=10,ysize=10
plot,findgen(10),title=dummy
```

```
pp={p:!p, x:!x, y:!y, z:!z, map:!map, w:0,i:0,j:0,k:0,pp:0}
pmap=replicate(pp,200)
defsysv,'!pmap',pmap

;so !pmap is filled with a bunch of plot-related dummy variables

return
end



;************************************************************ *
;************************************************************ *
;***********        PMAP        ***************************
;************************************************************ *
;************************************************************ *
pro sub_pmap,k,get=get,put=put
compile_opt hidden
;takes care of setting plot variables.....


if keyword_set(get) then begin  ;!pmap variables are loaded from plot
!pmap(k).p   = !p
!pmap[k].p.position =[!x.window[0],!y.window[0],!x.window[1],!y.window[1] ]
!pmap(k).x   = !x
!pmap(k).y   = !y
!pmap(k).z   = !z
!pmap(k).map = !map
!pmap(k).w   = !d.window
!pmap(k).pp  = !p.multi[0]
endif



if keyword_set(put) then begin  ;pmap variables are 'applied' to plot

wset,       !pmap[k].w
!p   =     !pmap[k].p
!x   =     !pmap[k].x
!y   =     !pmap[k].y
!z   =     !pmap[k].z
!map =     !pmap[k].map

endif
```

```
return
end


;*********************************************************
;*********************************************************
;**********       TRASH1      **************************
;*********************************************************
;*********************************************************



pro trash1,xxx

;define !pmap in a start up routine called start
start


window,0
!p.multi=[0,2,2]

plot,findgen(10)
plot,findgen(20)
;so, two plots on a page

sub_pmap,0,/get
;so the !p,!x,!y,!z and !map variables are put into !pmap[0] structure

print,'111111111111111111111111'
print,!p.multi
print,!pmap[0].p.multi
print,'111111111111111111111111'


;doing plots on a second window
!p.multi=[0,3,2]
;or say, !p.multi=[0,0,0], makes no difference in what you see in window 0
window,2
plot,findgen(30)
plot,findgen(40)

print,'222222222222222222222222'
print,!p.multi
;this shows !p.multi=[4,3,2,0,0]
print,'222222222222222222222222'


;going back to window 0 to make some more plots
```

```
sub_pmap,0,/put
;in theory, the !p,!x,!y!z and !map variables should be what they were after the second plot in
window 0

print,'333333333333333333333333333333'
print,!p.multi
;this shows !p.multi=[2,2,2,0,0], which is OK
print,'333333333333333333333333333333'


plot,findgen(25)*findgen(25)

;this plot should be in window 0, in the lower left hand corner, but its in the upper right

print,'444444444444444444444444444444'
print,!p.multi
;!p.multi is now [1,2,2,0,0], so it's advanced, so the next plot should be in the lower left
print,'444444444444444444444444444444'

plot,sqrt(findgen(25))
print,'555555555555555555555555555555'
print,!p.multi
;!p.multi[0] is now 0, even though plot was still in upper right
print,'555555555555555555555555555555'


go=''
read,'hit return to continue  ',go

plot,findgen(100)
print,'666666666666666666666666666666'
print,!p.multi
print,'666666666666666666666666666666'
;print,3,2,,2,0,0, so should have been in upper left???

return
end
```

---

Subject: Re: strange !P.multi[0] behavior??
Posted by David Fanning on Thu, 07 Nov 2013 23:36:47 GMT
View Forum Message <> Reply to Message

M Q writes:

> I created a structure, !pmap, that stores the !p, !x, !y, !z, and !map variables so that I could make some plots in one window, then open another window and make plots, and then go back to

the first window to make more plots, starting where I left off. As I recall, it worked well.
>
> However, it doesn't work now, which means either my memory is faulty or else I made some changes that I don't recall. I've included the relevant code below.
>
> The point is that even though the program sub_pmap recalls the correct !p structure, that doesn't seem to affect the placement of plots. This is clear when you run the program.
>
> I'm at a loss as to what might be going on here. This is a capability which I found quite useful in the past and which I find that I need again in a big way.

You can't set both !P.Multi and !P.Position without getting the total
chaos you are experiencing. :-)

Comment this line out in Sub_PMap and things will work better for you:

```
;  !pmap[k].p.position =[!x.window[0],!y.window[0],$
    !x.window[1],!y.window[1] ]
```

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

---

Subject: Re: strange !P.multi[0] behavior??
Posted by David Fanning on Thu, 07 Nov 2013 23:45:25 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> You can't set both !P.Multi and !P.Position without getting the total
> chaos you are experiencing. :-)
>
> Comment this line out in Sub_PMap and things will work better for you:
>
> ;  !pmap[k].p.position =[!x.window[0],!y.window[0],$
>     !x.window[1],!y.window[1] ]

By the way, you might find it MUCH easier to use cgLayout to do this. It
is a much more flexible way to set up multiple plots and a great deal
easier to use under circumstances like this.

http://www.idlcoyote.com/idldoc/cg/cglayout.html

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

---

Subject: Re: strange !P.multi[0] behavior??
Posted by astroboy.20000 on Fri, 08 Nov 2013 20:57:33 GMT
View Forum Message <> Reply to Message

Oh duh.

What was I thinking? Thanks David.

I'll have a look at cglayout. The main reason I came up with sub_pmap and some related routines was that I was doing an analysis with multiple windows open where one window might have one huge plot and four or five small plots around the edges, and another window would be completely different. So, there was a need to carry around the position information.

I'm looking forward to your next book. I hope it's not too much of a trial.

Mark

---