Subject: Problems with draw widget scrolling in IDL 5.0
Posted by Roberto Racca on Mon, 11 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

I have encountered a few puzzling quirks in the Draw widget under IDL 5.0 (first full release) running on a Windows 95 platform. They have to do with the scrolling of the viewport over a larger drawing area.

1) I need to resize the underlying drawing area at run time. I use the widget_control procedure with the parameters draw_xsize and draw_ysize. Everything works properly if the viewport is located at the top left corner of the image. Otherwise, IDL appears to 'lose track' of the location of the viewport and starts displaying bands of grey background at the edges of the image, or in worst cases no image at all. The effect is cumulative over repeated resizings, and there is no way of restoring the proper positioning.

2) As a stopgap measure, I tried to have IDL reposition the viewport to the top left corner of the image before doing the resizing, only to discover that using widget_control with the set_draw_view parameter works erratically and sometimes not at all. It seems that IDL cannot shift the viewport toward smaller origin values than the current ones.

I enclose below a sample file which generates a small draw widget application. Clicking on the 'Toggle size' button will change the draw area from having the same size as the viewport to being twice as big, and back again. If the viewport is repositioned when the draw area is enlarged, and the size is then toggled, the problem described in point 1) above appears quite obviously.

I would sincerely appreciate having someone's opinion on this. I very much need the resizing feature for a critical application, and I cannot find a work-around. Does anyone know whether the problem been addressed in 5.0.2?

Many thanks,
Roberto Racca

---------- SAMPLE CODE. CUT HERE ------------------------------

```
pro testdraw_event, event

widget_control, event.id, get_uvalue = tag
case tag of
  "EXIT" : widget_control, event.top, /destroy
  "TOGGLE" : begin
    erase, !p.background
```

```
    widget_control, event.top, get_uvalue = Draw
    geometry = widget_info(Draw, /geometry)
    if geometry.draw_xsize eq 200 then begin
      widget_control, Draw, draw_xsize = 400, draw_ysize = 400
      tvscl, dist(400)
    endif else begin
      widget_control, Draw, draw_xsize = 200, draw_ysize = 200
      tvscl, dist(200)
    endelse
  end
endcase

end


pro testdraw

Base1 = widget_base(title = "Test of draw widget", /column)
Base2 = widget_base(Base1, /row)
But1 = widget_button(Base2, value="Toggle size", uvalue = "TOGGLE")
But2 = widget_button(Base2, value="Exit", uvalue = "EXIT")
Draw = widget_draw(Base1, xsize=200, ysize=200, $
                x_scroll_size=200, y_scroll_size=200)

widget_control, Base1, set_uvalue = Draw
widget_control, Base1, /realize
loadct, 5
tvscl, dist(200)

xmanager, "testdraw", Base1

end
```

--------- END OF SAMPLE CODE. CUT HERE ----------

---

Subject: Re: Problems with draw widget scrolling in IDL 5.0
Posted by Roberto Racca on Tue, 12 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

I have modified the small demo app which I originally posted
to include the ability to test the scrolling of the viewport under
software control. As I indicated in that post, the horizontal
scrolling fails to work when one attempts to move the viewport
to the left of its current position by means of set_draw_view.
The vertical scrolling appears to have no such problem, except that
there is a slight inconsistency between the maximum value to which it
can be set in software (y=199 in the demo) and the value to which it

can be 'pushed' manually (y=200), which is also the initial state. To test the software-controlled scrolling, enter an X and a Y value in the boxes below the draw widget and click the 'Scroll to:' button.
    I have also heard from RSI support: apparently they knew already about the bugs but missed the boat on 5.0.2. For your information, the widget works perfectly in the UNIX (Linux) version.

Roberto

---------- SAMPLE CODE. CUT HERE ----------

```
pro testdraw_event, event

widget_control, event.top, get_uvalue = widgIDs
Draw = widgIDs(0)
Label = widgIDs(1)
Text1 = widgIDs(2)
Text2 = widgIDs(3)
widget_control, event.id, get_uvalue = tag
case tag of
  "EXIT" : begin
    widget_control, event.top, /destroy
    return
  end
  "TOGGLE" : begin
    erase, !p.background
    geometry = widget_info(Draw, /geometry)
    if geometry.draw_xsize eq 200 then begin
      widget_control, Draw, draw_xsize = 400, draw_ysize = 400
      tvscl, dist(400)
    endif else begin
      widget_control, Draw, draw_xsize = 200, draw_ysize = 200
      tvscl, dist(200)
    endelse
  end
  "DRAW" : ; viewport scroll event (do nothing here)
  "SCROLL" : begin
    widget_control, Text1, get_value = X_text
    widget_control, Text2, get_value = Y_text
    widget_control, Draw, set_draw_view = [fix(X_text),fix(Y_text)]
  end
endcase

; Display the current viewport coordinates
widget_control, Draw, get_draw_view = temp
widget_control, Label, set_value="Viewport at: " + $
                    strtrim(string(temp(0)),2) + ", " + $
                    strtrim(string(temp(1)),2)
```

```
end


pro testdraw

Base1 = widget_base(title="Test of draw widget", /column)
Base2 = widget_base(Base1, /row)
But1 = widget_button(Base2, value="Toggle size", uvalue="TOGGLE")
But2 = widget_button(Base2, value="Exit", uvalue="EXIT")
Draw = widget_draw(Base1, xsize=200, ysize=200, $
                   x_scroll_size=200, y_scroll_size=200, $
             uvalue="DRAW", /viewport_events)
Label = widget_label(Base1, value="Viewport a: 0, 0", $
                     /dynamic_resize, /align_left)
Base3 = widget_base(Base1, /row)
But3 = widget_button(Base3, value="Scroll to:", uvalue="SCROLL")
Text1 = widget_text(Base3, xsize=5, ysize=1, uvalue="SCROLL", $
                    /editable)
Text2 = widget_text(Base3, xsize=5, ysize=1, uvalue="SCROLL", $
                    /editable)

widget_control, Base1, set_uvalue = [Draw, Label, Text1, Text2]
widget_control, Base1, /realize
loadct, 5
tvscl, dist(200)

xmanager, "testdraw", Base1

end

--------- END OF SAMPLE CODE. CUT HERE ----------------------------

--
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Roberto Racca, Ph.D.
Vice President for Research and Development
JASCO Research Ltd, Victoria, B.C., CANADA
tel. +1.250.5441187  fax +1.250.5444916
rob@jasco.com  http://www.jasco.com

---

## Subject: Re: Problems with draw widget scrolling in IDL 5.0
Posted by davidf on Tue, 12 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Roberto Racca writes:

> David Fanning <davidf@dfanning.com> wrote:
>
>> As a work-around, what about making the draw widget as large
>> as you will possibly need and then restricting the viewport
>> scrolling in such a way as to keep the viewport over the
>> current image.
>
> Yes, I had tried the approach of the constantly large draw area. It
> works, of course, but the problem is that I can think of no way to
> 'restrict the scrolling' as you say. It is most disconcerting to the
> users if, having scrolled the viewport to some region far from the
> top left corner and then zoomed back, they are confronted with a
> blank screen and have to 'search' for the image. As I mentioned in my
> original post, you cannot reliably force a scroll in software using
> set_draw_view in widget_control. Any thoughts?

The problem with resizing draw widgets and the problems with
the Set_Draw_View keyword to Widget_Control are both known
problems. RSI engineers are currently researching the cause
of the problem, which I'm told has a high priority. They
expect to have the problem solved in the next release of IDL.

Information about this and other assorted behaviors in IDL 5
can be found on the IDL 5 Reports and Information page of
my web page.

Cheers,

David

-------------------------------------------------------------- -----
David Fanning, Ph.D.
Fanning Software Consulting
Customizable IDL Programming Courses
Phone: 970-221-0438  E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com

---

Subject: Re: Problems with draw widget scrolling in IDL 5.0
Posted by Roberto Racca on Tue, 12 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning <davidf@dfanning.com> wrote in article
<MPG.e5a1df767a35afd9896dd@news.frii.com>...

> As a work-around, what about making the draw widget as large
> as you will possibly need and then restricting the viewport
> scrolling in such a way as to keep the viewport over the

> current image.

Yes, I had tried the approach of the constantly large draw area. It works, of course, but the problem is that I can think of no way to 'restrict the scrolling' as you say. It is most disconcerting to the users if, having scrolled the viewport to some region far from the top left corner and then zoomed back, they are confronted with a blank screen and have to 'search' for the image. As I mentioned in my original post, you cannot reliably force a scroll in software using set_draw_view in widget_control. Any thoughts?

Thanks!
Roberto

--
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Roberto Racca, Ph.D.
Vice President for Research and Development
JASCO Research Ltd, Victoria, B.C., CANADA
tel. +1.250.5441187 fax +1.250.5444916
rob@jasco.com http://www.jasco.com

---

Subject: Re: Problems with draw widget scrolling in IDL 5.0
Posted by davidf on Tue, 12 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Roberto Racca writes:

> I have encountered a few puzzling quirks in the Draw widget under IDL
> 5.0 (first full release) running on a Windows 95 platform. They have
> to do with the scrolling of the viewport over a larger drawing area.
>
> 1) I need to resize the underlying drawing area at run time. I use
> the widget_control procedure with the parameters draw_xsize and
> draw_ysize. Everything works properly if the viewport is located at
> the top left corner of the image. Otherwise, IDL appears to 'lose
> track' of the location of the viewport and starts displaying bands of
> grey background at the edges of the image, or in worst cases no image
> at all. The effect is cumulative over repeated resizings, and there
> is no way of restoring the proper positioning.
>
> I would sincerely appreciate having someone's opinion on this. I very
> much need the resizing feature for a critical application, and I cannot
> find a work-around. Does anyone know whether the problem been
> addressed in 5.0.2?

The problem is still apparent in IDL 5.0.2 on my Windows NT machine.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Toggling the size in your example program after I move the scroll bars results in a draw widget that is about 1 pixel by 1 pixel. I'll add this to my IDL 5 Information page, which is currently under construction.

As a work-around, what about making the draw widget as large as you will possibly need and then restricting the viewport scrolling in such a way as to keep the viewport over the current image.

Cheers,

David

-------------------------------------------------------------- ----
David Fanning, Ph.D.
Fanning Software Consulting
Customizable IDL Programming Courses
Phone: 970-221-0438  E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com