Subject: Re: Arithmetic Errors Windows 64-bit Object Graphics
Posted by David Fanning on Fri, 11 Dec 2009 16:00:04 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> I had occasion this morning to run FSC_SURFACE on my
> new Windows 64-bit OS (IDL 7.1 and 7.1.2) and both
> throw NUMERIOUS floating illegal operand errors when
> I rotate the surface. Nothing like this happens in
> my old Windows 32-bit OS (IDL 7.1). Has anyone else
> noticed this?

OK, this turns out to be a problem with the RETAIN=2
keyword on the Draw Widget. For some reason, with
IDL maintaining the backing store on 64-bit Windows
versions, an error is thrown. I'll report it to ITTVIS.

Cheers,

David

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

Subject: Re: Arithmetic Errors Windows 64-bit Object Graphics
Posted by Kenneth P. Bowman on Fri, 11 Dec 2009 16:11:14 GMT
View Forum Message <> Reply to Message

In article <MPG.258c1a74d1df8daa989699@news.giganews.com>,
David Fanning <news@dfanning.com> wrote:

> OK, this turns out to be a problem with the RETAIN=2
> keyword on the Draw Widget. For some reason, with
> IDL maintaining the backing store on 64-bit Windows
> versions, an error is thrown. I'll report it to ITTVIS.

On Mac OS X, iTools windows spew floating point errors when
multiple monitors are involved. All I have to do is move the
mouse around in the window. It does not seem to matter whether
RETAIN is set to 2 or 1.

I though this was probably an X-Windows problem, but perhaps

it is IDL.

Ken

Subject: Re: Arithmetic Errors Windows 64-bit Object Graphics
Posted by David Fanning on Fri, 11 Dec 2009 16:17:12 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> OK, this turns out to be a problem with the RETAIN=2
> keyword on the Draw Widget. For some reason, with
> IDL maintaining the backing store on 64-bit Windows
> versions, an error is thrown. I'll report it to ITTVIS.

One more update. I can fix this problem in either of
two ways. I can use software rendering instead of
hardware rendering, which fixes it for *me* if not
for the rest of the world. Or, I can do my own backing
store by using EXPOSE events. This is how I used to do
all my object graphics programs and the code is already
in place to do this, so this is how I'll fix this.

I'm told this kind of exception is extremely common
with hardware rendering of object graphics code. So
much so, that a great deal of code exists that turns
off exception handling when calling the Draw method
on IDLgrWindows. (Perhaps the reason we don't see the
problem in iTools. :-)

```
PRO CallDrawMethodWithoutErrors, thisWindow, thisView

  exceptSave = !Except
  !Except = 0
  thisWindow-> Draw, thisView
  dummy = Check_Math()
  !Except = exceptSave

END
```

Cheers,

David

--

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

Subject: Re: Arithmetic Errors Windows 64-bit Object Graphics
Posted by David Fanning on Fri, 11 Dec 2009 16:38:15 GMT
View Forum Message <> Reply to Message

Kenneth P. Bowman writes:

> On Mac OS X, iTools windows spew floating point errors when
> multiple monitors are involved. All I have to do is move the
> mouse around in the window. It does not seem to matter whether
> RETAIN is set to 2 or 1.
>
> I though this was probably an X-Windows problem, but perhaps
> it is IDL.

Have you tried software rendering? This is the solution
I think I am going to adopt because I see I switched from
using EXPOSE events originally because of undocumented
changes in XManager that caused EXPOSE events to not be
handled correctly in blocking widgets. Sigh...

Now I am setting the RENDERER keyword to 0 on my
draw widget. That also seems to fix the problem for
me. I've left all the possible solution code in the
program so the user can choose his own poison.

I'll say this, that Python book is creeping closer and
closer to the place where I do all my work.

Cheers,

David

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

Subject: Re: Arithmetic Errors Windows 64-bit Object Graphics

Posted by Karl[1] on Fri, 11 Dec 2009 18:36:13 GMT
View Forum Message <> Reply to Message

On Dec 11, 10:17 am, David Fanning <n...@dfanning.com> wrote:
> David Fanning writes:
>> OK, this turns out to be a problem with the RETAIN=2
>> keyword on the Draw Widget. For some reason, with
>> IDL maintaining the backing store on 64-bit Windows
>> versions, an error is thrown. I'll report it to ITTVIS.
>
> One more update. I can fix this problem in either of
> two ways. I can use software rendering instead of
> hardware rendering, which fixes it for *me* if not
> for the rest of the world. Or, I can do my own backing
> store by using EXPOSE events. This is how I used to do
> all my object graphics programs and the code is already
> in place to do this, so this is how I'll fix this.
>
> I'm told this kind of exception is extremely common
> with hardware rendering of object graphics code. So
> much so, that a great deal of code exists that turns
> off exception handling when calling the Draw method
> on IDLgrWindows. (Perhaps the reason we don't see the
> problem in iTools. :-)
>
> PRO CallDrawMethodWithoutErrors, thisWindow, thisView
>
>   exceptSave = !Except
>   !Except = 0
>   thisWindow-> Draw, thisView
>   dummy = Check_Math()
>   !Except = exceptSave
>
> END
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

The floating point errors are likely coming from your Windows display driver, probably the OpenGL modules. It would be worth checking with the card vendor for an update.

Subject: Re: Arithmetic Errors Windows 64-bit Object Graphics
Posted by David Fanning on Fri, 11 Dec 2009 19:02:51 GMT
View Forum Message <> Reply to Message

Karl writes:

> The floating point errors are likely coming from your Windows display
> driver, probably the OpenGL modules. It would be worth checking with
> the card vendor for an update.

Well, these drivers can't be more than a week old, but
I'll check. ;-)

Cheers,

David

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

Subject: Re: Arithmetic Errors Windows 64-bit Object Graphics
Posted by Kenneth P. Bowman on Fri, 11 Dec 2009 19:56:11 GMT
View Forum Message <> Reply to Message

In article <MPG.258c234f2f5d9cd298969b@news.giganews.com>,
David Fanning <news@dfanning.com> wrote:

> Have you tried software rendering? This is the solution
> I think I am going to adopt because I see I switched from
> using EXPOSE events originally because of undocumented
> changes in XManager that caused EXPOSE events to not be
> handled correctly in blocking widgets. Sigh...
>
> Now I am setting the RENDERER keyword to 0 on my
> draw widget. That also seems to fix the problem for
> me. I've left all the possible solution code in the
> program so the user can choose his own poison.
>
> Cheers,
>
> David

I still see the problem when I do

  isurface, dist(50), renderer=1

but it only happens when using multiple monitors, so I think
it is a Mac X-Windows bug.

Also, I only do 3-D graphics when I want to interact with the plot
(rotate, etc.). 3-D is largely useless otherwise (at least for
my purposes). Hardware rendering is necessary to get reasonable
speeds for the complex volumes and surfaces that we render.

Cheers, Ken

Subject: Re: Arithmetic Errors Windows 64-bit Object Graphics
Posted by R.Bauer on Thu, 17 Dec 2009 08:05:03 GMT
View Forum Message <> Reply to Message

Kenneth P. Bowman schrieb:
> In article <MPG.258c234f2f5d9cd298969b@news.giganews.com>,
> David Fanning <news@dfanning.com> wrote:
>
>> Have you tried software rendering? This is the solution
>> I think I am going to adopt because I see I switched from
>> using EXPOSE events originally because of undocumented
>> changes in XManager that caused EXPOSE events to not be
>> handled correctly in blocking widgets. Sigh...
>>
>> Now I am setting the RENDERER keyword to 0 on my
>> draw widget. That also seems to fix the problem for
>> me. I've left all the possible solution code in the
>> program so the user can choose his own poison.
>>
>> Cheers,
>>
>> David
>
> I still see the problem when I do
>
>   isurface, dist(50), renderer=1
>
> but it only happens when using multiple monitors, so I think
> it is a Mac X-Windows bug.
>
> Also, I only do 3-D graphics when I want to interact with the plot
> (rotate, etc.). 3-D is largely useless otherwise (at least for
> my purposes). Hardware rendering is necessary to get reasonable

> speeds for the complex volumes and surfaces that we render.
>
> Cheers, Ken

OMG

that problem may not only related to graphics output functions.

please request on your bug report a unit test program which we can use
to verify this bug on other systems. And not only a fix.

cheers
Reimar

Subject: Re: Arithmetic Errors Windows 64-bit Object Graphics
Posted by Kenneth P. Bowman on Thu, 17 Dec 2009 21:04:53 GMT
View Forum Message <> Reply to Message

In article <hgcof3$3sok$1@zam602.zam.kfa-juelich.de>,
Reimar Bauer <R.Bauer@fz-juelich.de> wrote:

> Kenneth P. Bowman schrieb:
>> In article <MPG.258c234f2f5d9cd298969b@news.giganews.com>,
>> David Fanning <news@dfanning.com> wrote:
>>
>>> Have you tried software rendering? This is the solution
>>> I think I am going to adopt because I see I switched from
>>> using EXPOSE events originally because of undocumented
>>> changes in XManager that caused EXPOSE events to not be
>>> handled correctly in blocking widgets. Sigh...
>>>
>>> Now I am setting the RENDERER keyword to 0 on my
>>> draw widget. That also seems to fix the problem for
>>> me. I've left all the possible solution code in the
>>> program so the user can choose his own poison.
>>>
>>> Cheers,
>>>
>>> David
>>
>> I still see the problem when I do
>>
>> isurface, dist(50), renderer=1
>>
>> but it only happens when using multiple monitors, so I think
>> it is a Mac X-Windows bug.
>>

>> Also, I only do 3-D graphics when I want to interact with the plot
>> (rotate, etc.). 3-D is largely useless otherwise (at least for
>> my purposes). Hardware rendering is necessary to get reasonable
>> speeds for the complex volumes and surfaces that we render.
>>
>> Cheers, Ken
>
> OMG
>
> that problem may not only related to graphics output functions.
>
> please request on your bug report a unit test program which we can use
> to verify this bug on other systems. And not only a fix.
>
> cheers
> Reimar

Hi Reimar,

I'm not sure what you are asking.

All that is necessary to generate floating-point errors is

1. Have an external monitor attached to my MacBook Pro.

2. Run

   IDL> isurface, dist(50)

3. Rotate the resulting surface. If I drag the X-Window
   to the external monitor, it generates floating-point
   errors when I merely move the mouse around inside the window.

Ken
