Subject: the "real" screen size
Posted by wlandsman on Sun, 24 Oct 2010 22:02:02 GMT
View Forum Message <> Reply to Message

I often like to view images as big as possible on my monitor. So I get the screen size using device,get_screen_size=winsize, and open a window of this size using WINDOW, XSIZE=, YSIZE=. I then CONGRID() my (bigger) image down to this size and display it with TV.

After all these years, I now realize that this method results in significant truncation of the image on my Linux (Redhat) box (and smaller truncation on my Mac). Device,get_screen_size reports a screen size of 1600 x 1200 on my LInux box, but when I open a 1200 x 1200 window, I am actually only viewing the first 1115 pixels of the image in the Y direction, so I am missing more than 7% of the image, presumably due to pixels taken up by the window and taskbar margins

There was a thread a while back (http://tinyurl.com/2bssnfe) on using the exclude_Taskbar keyword in the IDLsysMonitorInfo object to get the "free" screen size. But this method seem more relevant to determining the available size for a widget GUI, and in any case, the exclude_Taskbar keyword is only available for Windows OS.

I can always introduce a fudge factor (i.e. subtract 85 pixels from the reported screen size) but does anyone know any IDL or X-window settings that might help?

Thanks, --Wayne

Subject: Re: the "real" screen size
Posted by wlandsman on Mon, 25 Oct 2010 16:09:24 GMT
View Forum Message <> Reply to Message

On Oct 25, 12:02 pm, David Fanning <n...@dfanning.com> wrote:
> wlandsman writes:
>> ;Note -- we can't use /PIXMAP so a new window will momentarily flash.
>
> Why can't you use a pixmap?
>

Because with /PIXMAP it appears that !D.X_size, !D.Y_size then match the original GET_SCREEN_SIZE() values. (This makes sense -- one can create images larger than the display with a pixmap, and one doens't have to worry about the area covered by a taskbar.) --Wayne

Subject: Re: the "real" screen size
Posted by David Fanning on Mon, 25 Oct 2010 16:15:29 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Why can't you use a pixmap?

Oh, never mind. I see. Darn, I hate flashing
windows about as much as I hate anything. :-(

Nothing speaks "non-commercial" more than these
kinds of solutions.

Are the LINUX and Mac numbers all over the board,
or could you do a reasonable "guess" and just
incorporate those into our ever-present fudge
factors?

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: the "real" screen size
Posted by Michael Galloy on Mon, 25 Oct 2010 17:32:28 GMT
View Forum Message <> Reply to Message

On 10/25/10 10:15 AM, David Fanning wrote:
> David Fanning writes:
>
>> Why can't you use a pixmap?
>
> Oh, never mind. I see. Darn, I hate flashing
> windows about as much as I hate anything. :-(
>
> Nothing speaks "non-commercial" more than these
> kinds of solutions.
>
> Are the LINUX and Mac numbers all over the board,
> or could you do a reasonable "guess" and just

> incorporate those into our ever-present fudge
> factors?

The EXCLUDE_TASKBAR is ignored on Macs, BUT it always returns the size
without the menu bar (basically its the equivalent of
EXCLUDE_TASKBAR=1 always):

For my work monitor (Mac OS X, 1680 x 1050), I get:

```
IDL> print, getPrimaryScreenSize()
       1680        1028
IDL> print, getPrimaryScreenSize(/exclude)
       1680        1028
```

This seems consistent from my home laptop: the menu bar is 22 pixels tall.

Mike
--
www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: the "real" screen size
Posted by wlandsman on Mon, 25 Oct 2010 18:09:49 GMT
View Forum Message <> Reply to Message

On Oct 25, 1:32 pm, mgalloy <mgal...@gmail.com> wrote:
>
>
> The EXCLUDE_TASKBAR is ignored on Macs, BUT it always returns the size
> *without* the menu bar (basically its the equivalent of
> EXCLUDE_TASKBAR=1 always):

I find this also -- so it looks like for Macs we can avoid the
flashing to get the maximum useable image area, and that a program
GetRealScreenSize() should have separate branches for Mac, Windows,
and Linux.

A couple of other notes:

1. The value of !D.Y_SIZE on my Linux box is intelligent, in that if
I hide the toolbar, I will get a larger value of !D.Y_SIZE, i.e. I
will be able to write a larger IDL image.
2. An alternative to device,get_screen_size = win is to use the
GET_SCREEN_SIZE() function in the ITTVIS library. Although the
change is not documented, this function was completely rewritten in
IDL 8.0 to use the IDLsysMonitorInfo object. However, it does not

accept the EXCLUDE_TASKBAR keyword. --Wayne

IDL> print,get_screen_size()
       1600   1200

Subject: Re: the "real" screen size
Posted by Michael Galloy on Tue, 26 Oct 2010 19:41:01 GMT
View Forum Message <> Reply to Message

On 10/25/10 12:09 PM, wlandsman wrote:
> On Oct 25, 1:32 pm, mgalloy<mgal...@gmail.com> wrote:
>>
>>
>> The EXCLUDE_TASKBAR is ignored on Macs, BUT it always returns the size
>> *without* the menu bar (basically its the equivalent of
>> EXCLUDE_TASKBAR=1 always):
>
> I find this also -- so it looks like for Macs we can avoid the
> flashing to get the maximum useable image area, and that a program
> GetRealScreenSize() should have separate branches for Mac, Windows,
> and Linux.
>
> A couple of other notes:
>
> 1. The value of !D.Y_SIZE on my Linux box is intelligent, in that if
> I hide the toolbar, I will get a larger value of !D.Y_SIZE, i.e. I
> will be able to write a larger IDL image.
> 2. An alternative to device,get_screen_size = win is to use the
> GET_SCREEN_SIZE() function in the ITTVIS library. Although the
> change is not documented, this function was completely rewritten in
> IDL 8.0 to use the IDLsysMonitorInfo object. However, it does not
> accept the EXCLUDE_TASKBAR keyword. --Wayne
>
> IDL> print,get_screen_size()
>   1600   1200
>

I think we are getting slightly different things with the Windows/Mac
vs. UNIX solutions (at least as of David's current MaxWindowSize
routine): Mac is returning the size of the available space to put a
window (the full size of a maximally sized window including the menubar)
and UNIX is returning the size of the graphics part of a maximally sized
window.

So the following isn't true for Macs (I don't know about Windows):

; To create a window of maximum size::

```
;       maxsize = MaxWindowSize()
;       Window, XSize=maxsize[0], YSize=maxsize[1], /Free
```

This creates a window that you can't see the bottom of because the menubar of the graphics window is "pushing" the window down a bit extra (remember there are two menubars when dealing with an X window on Mac OS X: the normal Mac menubar that is always there *and* a Windows-like menubar on each window).

If you want to create a maximally sized graphics window so that the above example code works, then on Mac I think it has to use the UNIX solution. Does this work on Windows? IDLsysMonitor::getRectangles with EXCLUDE_TASKBAR excludes the windows menubar too?

Mike
--
www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: the "real" screen size
Posted by David Fanning on Tue, 26 Oct 2010 20:03:45 GMT
View Forum Message <> Reply to Message

Mike Galloy writes:

> If you want to create a maximally sized graphics window so that the
> above example code works, then on Mac I think it has to use the UNIX
> solution.

Well, this is *extremely* confusing, which is why I'm trying to write it down. So, what do you think the Mac is reporting, then, when you do Get_Screen_Size()? You think this is the window *without* a title bar, but not taking into account the task bar, is that right?

I'm I correct that:

```
  Device, Get_Screen_Size=theSize
```

On a Mac, actually gives the screen or monitor size?

> Does this work on Windows? IDLsysMonitor::getRectangles with
> EXCLUDE_TASKBAR excludes the windows menubar too?

It would appear so, yes.

Thanks for your help with this. Not having a Mac, and
not knowing all these machine-specific nuances, is what
makes writing machine independent code so very, very
difficult!

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: the "real" screen size
Posted by Michael Galloy on Tue, 26 Oct 2010 20:39:04 GMT
View Forum Message <> Reply to Message

On 10/26/10 2:03 PM, David Fanning wrote:
> Mike Galloy writes:
>
>> If you want to create a maximally sized graphics window so that the
>> above example code works, then on Mac I think it has to use the UNIX
>> solution.
>
> Well, this is *extremely* confusing, which is why I'm trying
> to write it down. So, what do you think the Mac is reporting,
> then, when you do Get_Screen_Size()? You think this is the
> window *without* a title bar, but not taking into account the
> task bar, is that right?

It's the window without the menubar (the top one that is always there on
Mac OS X, sort of like the Windows taskbar that is always at the
bottom), but not accounting for the menubar on individual X windows.
It's just the "usable space" for the monitor. If you could create a
graphics window without a menubar, you could use all that space.

My screen resolution is 1680 x 1050:

```
IDL> print, get_screen_size()
        1680        1028
```

> I'm I correct that:
>
>   Device, Get_Screen_Size=theSize
>

> On a Mac, actually gives the screen or monitor size?

This, as well as using IDLsysMonitorInfo::getRectangles(), returns the
same thing:

IDL> Device, Get_Screen_Size=theSize
IDL> print, thesize
    1680   1028

>> Does this work on Windows? IDLsysMonitor::getRectangles with
>> EXCLUDE_TASKBAR excludes the windows menubar too?
>
> It would appear so, yes.
>
> Thanks for your help with this. Not having a Mac, and
> not knowing all these machine-specific nuances, is what
> makes writing machine independent code so very, very
> difficult!

Mike
--
www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: the "real" screen size
Posted by David Fanning on Tue, 26 Oct 2010 20:41:00 GMT
View Forum Message <> Reply to Message

Mike Galloy writes:

> My screen resolution is 1680 x 1050:
>
> IDL> print, get_screen_size()
>   1680   1028
>
>> I'm I correct that:
>>
>>   Device, Get_Screen_Size=theSize
>>
>> On a Mac, actually gives the screen or monitor size?
>
> This, as well as using IDLsysMonitorInfo::getRectangles(), returns the
> same thing:
>
> IDL> Device, Get_Screen_Size=theSize
> IDL> print, thesize

>   1680   1028

OK, thanks. Back to the drawing board. :-)

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: the "real" screen size
Posted by David Fanning on Wed, 27 Oct 2010 14:10:00 GMT
View Forum Message <> Reply to Message

Mike Galloy writes:

>>> If you want to create a maximally sized graphics window so that the
>>> above example code works, then on Mac I think it has to use the UNIX
>>> solution.

After further investigation overnight and this morning, I've
finally come to the conclusion that everyone using IDL on
a Mac should be shot. You're screwing up our lives! :-(

It turns out that our "UNIX" solution to finding the
"maximum" size window doesn't work for Macs. In fact,
NO solution works for Macs, except the tried and true
fudge factor solution.

Unless anyone can offer me hope, I'll change my article
appropriately. Maybe I'll program my MaxWindowSize program
to turn the fire alarm on if you call it from a Mac. :-)

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: the "real" screen size
Posted by ronn on Wed, 27 Oct 2010 21:38:58 GMT
View Forum Message <> Reply to Message

Hello All,

I may be missing something, but this slight modification to Dick's program
works on my Mac....

-Ronn

```
FUNCTION GetPrimaryScreenSize, Exclude_Taskbar=exclude_Taskbar

oMonInfo = Obj_New('IDLsysMonitorInfo')
rects = oMonInfo -> GetRectangles(Exclude_Taskbar=exclude_Taskbar)
pmi = oMonInfo -> GetPrimaryMonitorIndex()
Obj_Destroy, oMonInfo
if !d.name eq 'WIN' then begin
  Return, rects[[2, 3], pmi]              ; w & h of primary monitor avbl.
space
endif else begin
  return, rects[[2, 3], pmi] - [0,rects[1]]
endelse
END
```

Subject: Re: the "real" screen size
Posted by David Fanning on Wed, 27 Oct 2010 22:11:15 GMT
View Forum Message <> Reply to Message

ronn kling writes:

```
>  I may be missing something, but this slight modification to Dick's program
>  works on my Mac....
>
>  -Ronn
>
>  FUNCTION GetPrimaryScreenSize, Exclude_Taskbar=exclude_Taskbar
>
>  oMonInfo = Obj_New('IDLsysMonitorInfo')
>  rects = oMonInfo -> GetRectangles(Exclude_Taskbar=exclude_Taskbar)
>  pmi = oMonInfo -> GetPrimaryMonitorIndex()
>  Obj_Destroy, oMonInfo
>  if !d.name eq 'WIN' then begin
>    Return, rects[[2, 3], pmi]              ; w & h of primary monitor avbl.
>  space
>  endif else begin
>    return, rects[[2, 3], pmi] - [0,rects[1]]
```

> endelse
> END

I don't have a Macintosh, obviously, or I would have
a better idea of what I was talking about, probably.
I think this might indeed work, but for the wrong reasons. :-)

That is to say, the monitor size is wrong, by about 22 pixels,
and then when you subtract another 22 pixels from that wrong
value, you get (surprise!!) exactly the fudge factor I've
hardcoded for the "dock". So, in effect, you found the right
value by subtracting the a decent guess at the fudge factor
from the wrong monitor size value.

Humm. Yes, I guess that algorithm is as good as any! :-)

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: the "real" screen size
Posted by Kenneth P. Bowman on Thu, 28 Oct 2010 14:54:52 GMT
View Forum Message <> Reply to Message

In article <MPG.2731e0a4a3ee52b998980f@news.giganews.com>,
David Fanning <news@dfanning.com> wrote:

> Mike Galloy writes:
>
>>>> If you want to create a maximally sized graphics window so that the
>>>> above example code works, then on Mac I think it has to use the UNIX
>>>> solution.
>
> After further investigation overnight and this morning, I've
> finally come to the conclusion that everyone using IDL on
> a Mac should be shot. You're screwing up our lives! :-(

I think the solution is to run X-Windows on the Mac in full screen mode.

And, no, I am not serious, although there are times when I have resorted to doing that.

Ken
