Subject: Re: the "real" screen size
Posted by David Fanning on Mon, 25 Oct 2010 00:16:19 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> 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?

I run into this problem over and over again. Fudge
factors are the only thing I've ever known to help.
In fact, I'm not sure you don't need different fudge
factors for different flavors of UNIX. :-(

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 Carsten Lechte on Mon, 25 Oct 2010 08:56:35 GMT
View Forum Message <> Reply to Message

On 25/10/10 00:02, wlandsman wrote:
>
> 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.

At least on my linux box, if you do that, you can then examine the
size fields in the !D system variable to get the true size of your
window, which you can assume to be the maximum possible size and use
accordingly in your subsequent drawing operations.

Fudge factors are not a good idea, since it is up to the window
manager and the user's preferences how much screen real estate is
consumed by toolbars, window title bars, and other decorations.

chl

Subject: Re: the "real" screen size
Posted by David Fanning on Mon, 25 Oct 2010 12:19:08 GMT
View Forum Message <> Reply to Message

Carsten Lechte writes:

> Fudge factors are not a good idea, since it is up to the window
> manager and the user's preferences how much screen real estate is
> consumed by toolbars, window title bars, and other decorations.

I think everyone agrees fudge factors are a Bad Idea.
But, unfortunately, we write programs in the real
world, not a theoretical one. :-)

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 Carsten Lechte on Mon, 25 Oct 2010 12:42:35 GMT
View Forum Message <> Reply to Message

On 25/10/10 14:19, David Fanning wrote:
> I think everyone agrees fudge factors are a Bad Idea.
> But, unfortunately, we write programs in the real
> world, not a theoretical one. :-)

Yes, yes, but in this case, I meant that fudge factors are bad because
I think I have given a better alternative ;-)

chl

Subject: Re: the "real" screen size
Posted by lecacheux.alain on Mon, 25 Oct 2010 13:01:24 GMT
View Forum Message <> Reply to Message

On 25 oct, 00:02, wlandsman <wlands...@gmail.com> wrote:
> 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

In MS-Windows (I do not know in Unix/Linux), IDL window creation looks
like to be different in DG and new Graphics.
After Device,GET_SCREEN_SIZE=sz, the statement Window,/
FREE,XSIZE=sz[0],YSIZE=sz[1] creates a somewhat truncated window,
overlapped by system taskbar, while !Null=Window(DIMENSIONS=sz)
creates a well centered, not overlapped window.
In the second case, I guess that it would be possible to understand
what IDL is actually doing, by using Widget_WINDOW and Widget_INFO(/
GEOMETRY) or, likely better, by using ITOOLS armada, since new
graphics windows should derive from ITools...
Alx.

Subject: Re: the "real" screen size
Posted by David Fanning on Mon, 25 Oct 2010 13:10:35 GMT
View Forum Message <> Reply to Message

Carsten Lechte writes:

> Yes, yes, but in this case, I meant that fudge factors are bad because
> I think I have given a better alternative ;-)

Yes, which is what? I must not have gotten that post. Sorry.

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 wlandsman on Mon, 25 Oct 2010 13:26:44 GMT
View Forum Message <> Reply to Message

On Oct 25, 4:56 am, Carsten Lechte <c...@toppoint.de> wrote:

>
> At least on my linux box, if you do that, you can then examine the
> size fields in the !D system variable to get the true size of your
> window, which you can assume to be the maximum possible size and use
> accordingly in your subsequent drawing operations.

That works on both my Linux box and Mac-- thanks! (I could swear
that it didn't used to work, but I won't worry about that now.) --
Wayne

Subject: Re: the "real" screen size
Posted by David Fanning on Mon, 25 Oct 2010 13:32:03 GMT
View Forum Message <> Reply to Message

wlandsman writes:

>
> On Oct 25, 4:56 am, Carsten Lechte <c...@toppoint.de> wrote:
>
>>
>> At least on my linux box, if you do that, you can then examine the
>> size fields in the !D system variable to get the true size of your

>> window, which you can assume to be the maximum possible size and use
>> accordingly in your subsequent drawing operations.
>
> That works on both my Linux box and Mac-- thanks! (I could swear
> that it didn't used to work, but I won't worry about that now.) --

Ah, is this what Carsten means:

```
IDL> device, get_screen_size=s
IDL> print,s
   1280    1024
IDL> window, xsize=1280, ysize=1024
IDL> print, !d.x_size, !d.y_size
       1280        1024
```

If so, then you only need fudge factors for Windows. ;-)

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 Mon, 25 Oct 2010 13:36:33 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Ah, is this what Carsten means:
>
> IDL> device, get_screen_size=s
> IDL> print,s
>   1280   1024
> IDL> window, xsize=1280, ysize=1024
> IDL> print, !d.x_size, !d.y_size
>   1280   1024
>
> If so, then you only need fudge factors for Windows. ;-)

Ah, yes. I just fired up my Linux machine. These size
variables do contain the real window size. Now, if we
just could find a cross-platform solution we would be

golden!

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 lecacheux.alain on Mon, 25 Oct 2010 13:54:09 GMT
View Forum Message <> Reply to Message

On 25 oct, 15:32, David Fanning <n...@dfanning.com> wrote:
> IDL> device, get_screen_size=s
> IDL> print,s
> 1280 1024
> IDL> window, xsize=1280, ysize=1024
> IDL> print, !d.x_size, !d.y_size
> 1280 1024
>
On Linux, after the window creation, do you see only a black screen ?
If not, as it is the case on my Windows box (I can see window frames,
system taskbar, etc...), !D.X_SIZE and !D.Y_SIZE does not give you the
drawing size for your image.
alx.

Subject: Re: the "real" screen size
Posted by Michael Galloy on Mon, 25 Oct 2010 13:58:19 GMT
View Forum Message <> Reply to Message

On 10/25/10 7:36 am, David Fanning wrote:
> David Fanning writes:
>
>> Ah, is this what Carsten means:
>>
>> IDL> device, get_screen_size=s
>> IDL> print,s
>> 1280 1024
>> IDL> window, xsize=1280, ysize=1024
>> IDL> print, !d.x_size, !d.y_size

>>   1280   1024
>>
>> If so, then you only need fudge factors for Windows. ;-)
>
> Ah, yes. I just fired up my Linux machine. These size
> variables do contain the real window size. Now, if we
> just could find a cross-platform solution we would be
> golden!

IDL> print, !version
{ x86_64 darwin unix Mac OS X 8.0 Jun 17 2010   64   64}
IDL> device, get_screen_size=s
IDL> print, s
    1440   878
IDL> print, !d.x_size, !d.y_size
         720   439

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

Subject: Re: the "real" screen size
Posted by Carsten Lechte on Mon, 25 Oct 2010 14:07:39 GMT
View Forum Message <> Reply to Message

On 25/10/10 15:32, David Fanning wrote:
> Ah, is this what Carsten means:
>
> IDL> device, get_screen_size=s
> IDL> print,s
>   1280   1024
> IDL> window, xsize=1280, ysize=1024
> IDL> print, !d.x_size, !d.y_size
>   1280   1024

Does that window then have a usable graphics area of 1280x1024, i.e. the window including borders and title bar is a bit larger than your screen? Or do the sizes in !D lie to you and the usable window size is really smaller than that?

In that case, the assumption is that the windowing system will automatically give you a properly maximized window if you request a size that is too large to be displayed AND the actual size is correctly reflected in the !D system variable.

We have already found that this assumption does not seem to hold on windows. And one can probably find a window manager on unix for which it does not hold either.

chl

Subject: Re: the "real" screen size
Posted by David Fanning on Mon, 25 Oct 2010 14:17:23 GMT
View Forum Message <> Reply to Message

alx writes:

> On Linux, after the window creation, do you see only a black screen ?
> If not, as it is the case on my Windows box (I can see window frames,
> system taskbar, etc...), !D.X_SIZE and !D.Y_SIZE does not give you the
> drawing size for your image.
> alx.

I think everyone is getting a bit confused here. Let me see if I can summarize. The problem we are trying to solve is that we want to create a window on the display that is as big as possible, without being obscured by window decorations, borders, etc. The question we are trying to ask is "How can we find the size of that window in a machine-independent way?"

One would think that the Get_Screen_Size keyword to the Device command would work:

```
IDL> Device, Get_Screen_Size=theSize
IDL> Print, thesize
     1280  1024
```

The problem is, if you make a window of this size, it is "too big" for the window. It is obscured.

Carsten's solution for LINUX is to make a window this size anyway (probably as a pixmap, I would assume) and then examine the variables !D.X_Size and !D.Y_Size. These will contain the sizes you are looking for.

UNIX:

```
IDL> Window, XSIZE=theSize[0], YSIZE=theSize[1]
IDL> Print, !D.X_Size, !D.Y_Size
      1278   944
```

Alas, this doesn't work for Windows computers:

```
  WINDOWS:
    IDL> Window, XSIZE=theSize[0], YSIZE=theSize[1]
    IDL> Print, !D.X_Size, !D.Y_Size
         1280  1024
```

So, we are still looking for a machine-independent solution.
At the moment, however, we only have to calculate fudge
factors for Windows machines, which is an improvement.
These fundge factor only depend on which version of Windows
you are using, how you have configured your machine, and
other factors too numerous to mention. :-)

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 Mon, 25 Oct 2010 14:18:31 GMT
View Forum Message <> Reply to Message

Michael Galloy writes:

```
>  IDL> print, !version
>  { x86_64 darwin unix Mac OS X 8.0 Jun 17 2010      64      64}
>  IDL> device, get_screen_size=s
>  IDL> print, s
>       1440     878
>  IDL> print, !d.x_size, !d.y_size
>            720         439
```

Yes, this is the part I missed, too, at first. You
have to make a large window first.

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 wlandsman on Mon, 25 Oct 2010 15:28:26 GMT
View Forum Message <> Reply to Message

On Oct 25, 10:17 am, David Fanning <n...@dfanning.com> wrote:
> So, we are still looking for a machine-independent solution.
> At the moment, however, we only have to calculate fudge
> factors for Windows machines, which is an improvement.
> These fudge factor only depend on which version of Windows
> you are using, how you have configured your machine, and
> other factors too numerous to mention. :-)

Does the function below originally written by Dick Jackson work for
Windows? (I don't have a Windows machine to test it.)

--Wayne

```
;;    Handy function for getting screen size of primary monitor,
optionally
;;    excluding the taskbar (Windows only)
;;
;;    Example:
;;    freeSize = GetPrimaryScreenSize(/Exclude_Taskbar)
;;    Print, freeSize
;;          1280        946

FUNCTION GetPrimaryScreenSize, Exclude_Taskbar=exclude_Taskbar

oMonInfo = Obj_New('IDLsysMonitorInfo')
rects = oMonInfo -> GetRectangles(Exclude_Taskbar=exclude_Taskbar)
pmi = oMonInfo -> GetPrimaryMonitorIndex()
Obj_Destroy, oMonInfo
Return, rects[[2, 3], pmi]        ; w & h of primary monitor avbl.
space

END
```

Subject: Re: the "real" screen size
Posted by David Fanning on Mon, 25 Oct 2010 15:33:11 GMT
View Forum Message <> Reply to Message

wlandsman writes:

> Does the function below originally written by Dick Jackson work for
> Windows? (I don't have a Windows machine to test it.)

Well, not really.

```
IDL> print, getprimaryscreensize()
      1280        1024
IDL> device, get_screen_size=s & print, s
   1280    1024
```

The monitor *is* that big, but windows created that
size fall behind the Windows start bar, etc.

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 Mon, 25 Oct 2010 15:41:35 GMT
View Forum Message <> Reply to Message

wlandsman writes:

> Does the function below originally written by Dick Jackson work for
> Windows? (I don't have a Windows machine to test it.)

Oh, wait! Maybe this does work. Too much multi-tasking going
on this morning. I didn't see that keyword:

```
IDL> print, getprimaryscreensize(/exclude)
      1280        994
```

Yes, this appears to work. I presume it works on LINUX
and Macs, too? Could be our solution! Thanks!

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 wlandsman on Mon, 25 Oct 2010 16:00:02 GMT
View Forum Message <> Reply to Message

On Oct 25, 11:41 am, David Fanning <n...@dfanning.com> wrote:
> wlandsman writes:

> Yes, this appears to work. I presume it works on LINUX
> and Macs, too?
>

No, the /Exclude_taskbar keyword is only for Windows OS. So our
function to get the maximum display area now looks like this:

```
FUNCTION GetRealScreenSize

if !VERSION.OS EQ 'Windows' then begin

oMonInfo = Obj_New('IDLsysMonitorInfo')
rects = oMonInfo -> GetRectangles(/Exclude_Taskbar)
pmi = oMonInfo -> GetPrimaryMonitorIndex()
Obj_Destroy, oMonInfo
Return, rects[[2, 3], pmi]        ; w & h of primary monitor avbl.

endif else begin

;Note -- we can't use /PIXMAP so a new window will momentarily flash.

device,get_screen_size=win
window,xsize= win[0],ysize=win[1],/free
win = [!D.X_SIZE,!D.Y_SIZE]
wdelete,!D.Window

return,win
```

END

Subject: Re: the "real" screen size
Posted by David Fanning on Mon, 25 Oct 2010 16:02:48 GMT
View Forum Message <> Reply to Message

wlandsman writes:

> ;Note -- we can't use /PIXMAP so a new window will momentarily flash.

Why can't you use a pixmap?

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.")