Subject: Re: "free" screen size
Posted by David Fanning on Sat, 25 Nov 2006 14:13:35 GMT
View Forum Message <> Reply to Message

lory writes:

> How can I find the "free" screen size ? I mean the size of the area
> available to display GUIs, something like the dimensions returned by
> GET_SCREEN_SIZE but subtracting the area dedicated to the task bar, if
> it is displayed.

I'm afraid you have to find this empirically. :-(

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: "free" screen size
Posted by Andrew Cool on Sun, 26 Nov 2006 08:54:36 GMT
View Forum Message <> Reply to Message

lory wrote:
> How can I find the "free" screen size ? I mean the size of the area
> available to display GUIs, something like the dimensions returned by
> GET_SCREEN_SIZE but subtracting the area dedicated to the task bar, if
> it is displayed.
>
> Lory

Lory,

If you're talking MS Windows, then I can let you have a small
executable that I knocked up
in another language that returns this info.

Using the executable in IDL, it's as easy as :-

size = window_size()

Let me know if you want a copy.

Cheers,

Andrew Cool

Subject: Re: "free" screen size
Posted by David Fanning on Sun, 26 Nov 2006 14:40:45 GMT
View Forum Message <> Reply to Message

Andrew Cool writes:

> If you're talking MS Windows, then I can let you have a small
> executable that I knocked up
> in another language that returns this info.
>
> Using the executable in IDL, it's as easy as :-
>
> size = window_size()
>
>
> Let me know if you want a copy.

Alright, it's the Cool Special Kludge, written
after a long day filled with too many prawns on the barbie
(not to mention too many beers!), but I have to admit
it *does* work. :-)

And--a very nice feature for something this elegant--it
also keeps track of whether the user is auto-hiding the
task bar or not.

Cheers,

David

P.S. I hope you are passing the secret to this along to the
good folks at ITTVIS, Andrew. They could use something like this.
Hold out for the big bucks, though. Or, at the very lease, their
babe magnet of a T-Shirt. :-)

--
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: "free" screen size
Posted by Dick Jackson on Sun, 26 Nov 2006 16:27:04 GMT
View Forum Message <> Reply to Message

"lory" <lore2323@virgilio.it> wrote in message
news:1164462367.533281.304210@45g2000cws.googlegroups.com...
> How can I find the "free" screen size ? I mean the size of the area
> available to display GUIs, something like the dimensions returned by
> GET_SCREEN_SIZE but subtracting the area dedicated to the task bar, if
> it is displayed.
>
> Lory

In IDL 6.3, multiple monitor support was added, along with an option for excluding taskbar size on Windows. I wrote this function to get at the desired info easily:

```
;;    Handy function for getting screen size of primary monitor, optionally
;;    excluding the taskbar
;;
;;    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
```

Cheers,
-Dick

--
Dick Jackson Software Consulting http://www.d-jackson.com
Victoria, BC, Canada +1-250-220-6117 dick@d-jackson.com

Subject: Re: "free" screen size
Posted by Andrew Cool on Sun, 26 Nov 2006 21:47:02 GMT
View Forum Message <> Reply to Message

Dick Jackson wrote:
> "lory" <lore2323@virgilio.it> wrote in message
> news:1164462367.533281.304210@45g2000cws.googlegroups.com...
>> How can I find the "free" screen size ? I mean the size of the area
>> available to display GUIs, something like the dimensions returned by
>> GET_SCREEN_SIZE but subtracting the area dedicated to the task bar, if
>> it is displayed.
>>
>> Lory
>
> In IDL 6.3, multiple monitor support was added, along with an option for
> excluding taskbar size on Windows. I wrote this function to get at the
> desired info easily:
>
>
> ;; Handy function for getting screen size of primary monitor, optionally
> ;; excluding the taskbar
> ;;
> ;; 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
>
>
> Cheers,
> -Dick
>

Crikey! I pestered RSI/IITVIS to make this info available - but I
didn't know they'd
actually done it!

Way better than my kludge... ;-)

Andrew

Subject: Re: "free" screen size
Posted by Andrew Cool on Sun, 26 Nov 2006 22:08:52 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Andrew Cool writes:
>
>> If you're talking MS Windows, then I can let you have a small
>> executable that I knocked up
>> in another language that returns this info.
>>
>> Using the executable in IDL, it's as easy as :-
>>
>> size = window_size()
>>
>>
>> Let me know if you want a copy.
>
> Alright, it's the Cool Special Kludge, written
> after a long day filled with too many prawns on the barbie
> (not to mention too many beers!), but I have to admit
> it *does* work. :-)
>
> And--a very nice feature for something this elegant--it
> *also* keeps track of whether the user is auto-hiding the
> task bar or not.
>
> Cheers,
>
> David
>
> P.S. I hope you are passing the secret to this along to the
> good folks at ITTVIS, Andrew. They could use something like this.
> Hold out for the big bucks, though. Or, at the very lease, their
> babe magnet of a T-Shirt. :-)

Apart from the fact that Dick has pointed out that this window info is
already in 6.3,
I'll pass on the babe magnet T-Shirt - I've already got 3 babes living
under my roof,
4 if you include Lucy the Whippet, and I get out-voted enough as it is.

Andrew

Subject: Re: "free" screen size
Posted by David Fanning on Mon, 27 Nov 2006 23:10:23 GMT

Dick Jackson writes:

```
>  In IDL 6.3, multiple monitor support was added, along with an option for
>  excluding taskbar size on Windows. I wrote this function to get at the
>  desired info easily:
>
>
>  ;;    Handy function for getting screen size of primary monitor, optionally
>  ;;    excluding the taskbar
>  ;;
>  ;;    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
```

In my testing here, I find that Andrew's WINDOW_SIZE function
reports a window size of 1272 by 969 on my 1280 by 1024 display.
This accounts for the taskbar AND window decoration for the TLB,
including the window title bar.

Dick's GetPrimaryScreenSize reports 1280 by 996, which does NOT
include any window decoration (including the window title bar).
Moreover, the /EXCLUDE_TASKBAR keyword only works if the taskbar
property "keep taskbar on top of other windows" is set. If this
property is not set, then the entire window size is returned.
This is identical to what is returned in Get_Window_Size(). That
is to say, not as useful as it could be.

Bottom line, although Andrew's solution is a bit of a kludge, it
does seem to return more useful information if you have the
practical purpose of building a widget program in mind.

Cheers,

David
--
David Fanning, Ph.D.

Subject: Re: "free" screen size
Posted by Dick Jackson on Wed, 29 Nov 2006 19:16:46 GMT
View Forum Message <> Reply to Message

"David Fanning" <news@dfanning.com> wrote in message
news:MPG.1fd51e307f237a80989de9@news.frii.com...
> Dick Jackson writes:
>
>> In IDL 6.3, multiple monitor support was added, along with an option for
>> excluding taskbar size on Windows. I wrote this function to get at the
>> desired info easily:
>>
>>
>> ;; Handy function for getting screen size of primary monitor,
>> optionally
>> ;; excluding the taskbar
>> ;;
>> ;; 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
>
> In my testing here, I find that Andrew's WINDOW_SIZE function
> reports a window size of 1272 by 969 on my 1280 by 1024 display.
> This accounts for the taskbar AND window decoration for the TLB,
> including the window title bar.
>
> Dick's GetPrimaryScreenSize reports 1280 by 996, which does NOT
> include any window decoration (including the window title bar).
> Moreover, the /EXCLUDE_TASKBAR keyword only works if the taskbar
> property "keep taskbar on top of other windows" is set. If this
> property is not set, then the entire window size is returned.
> This is identical to what is returned in Get_Window_Size(). That

> is to say, not as useful as it could be.
>
> Bottom line, although Andrew's solution is a bit of a kludge, it
> does seem to return more useful information if you have the
> practical purpose of building a widget program in mind.

Just for fun, I'll quote the original request:

lory writes:

> How can I find the "free" screen size ? I mean the size of the area
> available to display GUIs, something like the dimensions returned by
> GET_SCREEN_SIZE but subtracting the area dedicated to the task bar, if
> it is displayed.

I don't see any mention of how *many* windows lory might want to put on the screen, so I think I covered the request with the routine above. I think that if the taskbar can be covered up by our window, then it's reasonable to report full screen size as being available. I do have one quibble with oMonInfo -> GetRectangles(), that if I have "Auto-hide the taskbar" set, the two rows of pixels which are *not* available to an IDL window are being counted as available.

Now, if you want to know about setting up windows on the screen precisely, this is what I use. TLBWindowDressing returns the width of the frame and title bar (actually the XY offset of the first widget you would add to a base) for a window with any TLB_Frame_Attr you wish to pass it. The size available to widgets inside a single window that will fill the screen can be found as follows:

```
scrSize = getprimaryscreensize()
wd=tlbwindowdressing()
xAvbl = scrSize[0] - 2*wd.frameWidth
yAvbl = scrSize[1] - wd.yoffset+wd.frameWidth
tlb=widget_base(space=0,xpad=0,ypad=0,/row)
wdraw=widget_draw(tlb,xsize=xAvbl,ysize=yAvbl)
```

And if you need to lay out several windows to fill the screen, you can use these numbers to figure that out, too.

```
;-----

; TLBWindowDressing
;+
; Returns a structure describing this window system's top-level base window
; attributes. Currently assumes that the title bar is on top and that
; the other three sides have equal frame width.
```

```
; @keyword  _Extra {in}{optional}
;          Extra keyword parameters to pass to the Widget_Base() call

; @returns  A structure of this form:
; <code>
; <br> {TLBWindowDressing, $
; <br> FrameWidth:value, $ ; Pixels used on all sides for window frame
; <br> XOffset:value, $    ; X offset of top-left pixel of contained widgets
; <br> YOffset:value }     ; Y offset of top-left pixel of contained widgets
; </code>

; @restrictions Menubar size is not handled, as the Geometry seems incorrect
;               for a base with a menubar.

; @examples
; <code> IDL> wd = TLBWindowDressing()
; <br>   IDL> Help, wd, /Structure
; <br>   ** Structure TLBWINDOWDRESSING, 3 tags, length=12, data length=12:
; <br>      FRAMEWIDTH      LONG              4
; <br>      XOFFSET         LONG              4
; <br>      YOFFSET         LONG             23
; <br>   IDL> wd = TLBWindowDressing(TLB_Frame_Attr=4) ; Suppress title bar
; <br>   IDL> Help, wd, /Structure
; <br>   ** Structure TLBWINDOWDRESSING, 3 tags, length=12, data length=12:
; <br>      FRAMEWIDTH      LONG              4
; <br>      XOFFSET         LONG              4
; <br>      YOFFSET         LONG              4
; </code>

; @author   Dick Jackson Software Consulting, www.d-jackson.com
;-

FUNCTION TLBWindowDressing, _Extra=extra

;; wTLB = Widget_Base(MBar=wMBar)   ; Could not get dependable sizes with
MBar
;; wMenu = Widget_Button(wMBar, Value='', /Menu)

wTLB = Widget_Base(_Extra=extra)
geom = Widget_Info(wTLB, /Geometry)
Widget_Control, wTLB, /Destroy
frameWidth = Long(geom.xSize)/2
Return, {TLBWindowDressing, $
        FrameWidth:frameWidth, $
        XOffset:frameWidth, $
        YOffset:Long(geom.ySize)-frameWidth}

END
```

;-----

Please note: I have only tested this on Windows, so any fixes for Unix are welcome.

Cheers,
-Dick

--
Dick Jackson Software Consulting http://www.d-jackson.com
Victoria, BC, Canada +1-250-220-6117 dick@d-jackson.com
