Chapter 14 Windows & Grafports

From:Tim Maroney
Subject: Re: Checking the validity of a refCon (window refcon)

In article <17662@dartvax.Dartmouth.EDU> ari@eleazar.dartmouth.edu (Ari Halberstadt) writes:

> >Here's a question for you folks out on NetLand. I'm trying
> >to figure out, with a reasonable degree of success, if the refCon
> >field of a window contains a valid handle. This must be determined
> >by a driver, which is separate from the the program which actually
> >created the window.

But it *is* your program, right? All the windows whose refCons you wil be looking for (if not at) were created by an application you wrote? Then probably the best check involves a little-known feature of the WindowRecord. You can put any value of 8 or greater into your own application's windows' windowKind field (IM I-276). Pick some arbitrary large number and the collisions with other applications using the same trick become unlikely. If your first check is this, then you should still do some other checks, but they're really just formalities. You will almost certainly never find another application which both uses this obscure trick and picks the same number for its windowKinds.

I'm kind of confused about when it is you expect to be looking at other windows. You should be able to check whether it's your application running in a variety of ways. You can check for the presence of your signature resource; you can have the application call your driver to tell it when the application is running, or even when it creates or destroys a window; you can install a 'STR ' with a particular ID that contains some distinctive string; and so on.

> >The tests currently performed are:
> >
> >Boolean ValidHandle(Handle hndl)
> >{
> > hndl = (Handle) GetWRefCon(window); /* copied here for brevity */
> > if (! hndl)
> > return(false);
> > if ((long) hndl & 1L) /* check for odd address */
> > return(false);
> > if (GetHandleSize(hndl) != expected_size)
> > return(false);
> >}

This looks mostly OK; it wouldn't be a bad idea to combine this kind of testing with one of the other kinds I mentioned. I do have a problem with the GetHandleSize. It could conceivably be called on a non-handle; the "hndl & 1" test doesn't establish that it's a handle. For instance, suppose the window belongs to an application that sticks the number 0xfffffffe into the refCon. Boom. You should check that the handle is even, then that it is a pointer into the application heap, then that what it points at is also an even pointer into the application heap, and then dereference the pointer to find out whether the longword at it is equal to some distinctive value you've stashed in all your refcon records at the beginning.

> >Actually, I've got three questions.
> >1. Is it ok to check for odd adresses on a 680x0? I assume Apple
> >wouldn't write NewHandle to create something on an odd address, since
> >access times are slower.

NewHandle will always create it on an even address, regardless of machine. Heaps are divided into blocks which are size-rounded up to even numbers (IM II-25).

> >2. I would like to check that the handle is within a reasonable area
> >of memory, i.e., within the application heap. However, since the
> >application may have several heap zones, what is the correct way to
> >check this? Is it sufficient to use ApplicZone and GetApplLimit?

Sure. Be sure to convert them to unsigned longwords and call StripAddress on them before you do your arithmetic. The same goes for the pointers you're checking. Unless you do the (in my opinion, dreadful) trick of putting a heap on the stack, all your heaps will fall within the application heap. (You could also put a heap in temporary memory, but then it's not really temporary, is it?)

Windows & Grafports

>3. Any more ideas for useful checks? I've already got a special
>long int stuck at the head of the place where the correct handle

>would point to, and I've thought of making sure the handle is
>below RomBase.

This shouldn't hurt, but it's redundant if you're already checking against the application heap.

>--
>Ari Halberstadt '91, "Long live succinct signatures"
--
Tim Maroney, Mac Software Consultant, sun!hoptoad!tim, tim@toad.com

●●●

From: pepke@loligo (Eric Pepke)
Subject: Re: Floating Windows

In article <18786@bellcore.bellcore.com> sdh@flash.UUCP (Stephen D Hawley) writes:

>I need a pointer for making windows that float on top of the rest of the
>world no matter what. I tried setting GostWindow to the window I wanted
>on top, but that doesn't work. I want the kind of windows in FullPaint.
>
>Do I have to dick around with the window list pulling them in front
>constantly?

April 1988 MacTutor had a bit on this. I wouldn't do it exactly the way they did (and, in fact, I didn't) because, for one thing, Apple people have suggested that it would probably break. Most of the reason for this seems to have to do with accessing the Window Manager port directly to change the region where the drag window rectangle shows.

There is no real trick, just a lot of special cases. Don't use GhostWindow. The first order solution is to keep a floating window on the top, put it on the top and never allow another window to come in front of it. This is relatively easy; don't use SelectWindow on a document window, but rather activate the window and bring it to just behind the rearmost floating window. The trickiness comes with desk accessory windows. You want the floating windows to be in front of all the application windows, but not in front of the desk accessory windows.

I found that the easiest way to solve the problem was to think about all the special cases in the context of the structure of my main loop. What should happen when you bring up a desk accessory in front of the floating windows? What then should happen when you bring up a document window in front of that? Do you disallow that, as HyperCard does, or do you do it somewhat more cleverly? What window should be highlighted as active? (HyperCard does it stupidly.)

Eric Pepke INTERNET: pepke@gw.scri.fsu.edu

●●●

From: murat@farcomp.UUCP (Murat Konar)
Subject: Re: How do you find a window's location?

In article <3258@usceast.UUCP> jwwalker@usceast.UUCP (Jim Walker) writes:

>I know how to set a window's location with MoveWindow and
>SizeWindow, but how do you *find* a window's location? If the
>window is visible, I can use
> (**((WindowPeek)window_ptr)->contRgn).rgnBBox
>but that seems to contain garbage when the window is invisible.
>Yet the Window Manager somehow keeps track of the locations of
>invisible windows.

Here's how I do it:
1) Save the ccurrent port
2) SetPort to the window in question

3) Grab a copy of its port rect
4) Do a LocalToGlobal on its topLeft and botRight
5) Restore the current port

VOILA!

You now have the window's location in global coordinates!

Murat N. Konar

●●●

From: wdh@well.sf.ca.us (Bill Hofmann)
Subject: Re: How do you find a window's location?
Summary: GlobalToLocal the topLeft of the portRect

In article <22352@dartvax.Dartmouth.EDU> mjm@eleazar.dartmouth.edu (Michael McClennen) writes:

> >jwwalker@usceast.UUCP (Jim Walker) writes:
> >>I know how to set a window's location with MoveWindow and
> >>SizeWindow, but how do you *find* a window's location? If the
> >>window is visible, I can use
> >> (**((WindowPeek)window_ptr)->contRgn).rgnBBox
> >>but that seems to contain garbage when the window is invisible.
> >>Yet the Window Manager somehow keeps track of the locations of
> >>invisible windows.
> >How about checking the grafPort's portRect?

The portRect of a grafPort is the top,left,bottom,right of the window in LOCAL coordinates, that is, the coordinate system of the window. (Note that the top,left need not be 0,0). What you want is the position of the top,left in GLOBAL coordinates. So, looking at Inside Mac Vol I-193:

```
Point getWindowLocation(WindowPtr wp)
{
    Point  where;
    GrafPtr   curPort;

    where.v = wp->portRect.top;
    where.h = wp->portRect.left;
    GetPort(&curPort);
    SetPort(wp);
    GlobalToLocal(&where);
    SetPort(curPort);
    return(where);
}
```

If you want to include the title bar, which is above and to the left of the content, you'll have to look at the strucRgn of the window.

-Bill Hofmann

●●●

From: amanda@mermaid.intercon.com (Amanda Walker)
Subject: Re: How do you find a window's location?

> > jwwalker@usceast.UUCP (Jim Walker) writes:
> > >how do you *find* a window's location?

In article <22352@dartvax.Dartmouth.EDU>, mjm@eleazar.dartmouth.edu (Michael McClennen) writes:
> > How about checking the grafPort's portRect?

The portRect is in local coordinates. What I do is SetPort to the Window, make a copy of the portRect, and call LocalToGlobal on the corners. If you really want to be studly, you should also save & restore the zoomed state...

--

Amanda Walker

●●●