

---

Subject: Re: Expose Events (easy questions)

Posted by [David Fanning](#) on Wed, 03 Apr 2002 17:09:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Ted Cary (tedcary@yahoo.com) writes:

- > Anyone with experience using scrollable, resizable draw widgets and RETAIN=0
- > should be able to answer these easily.

Leaves me out, sorry. :-(

- > What is contained in the X and Y fields of the structure generated by a draw
- > widget expose event ?

You've got me. The documentation is...well...the documentation. :-(

- > The APP\_SCROLL draw widget is in a resizable base. The expose events are
- > generated when the base is resized

I don't understand how this can happen. Can you provide more details about this?

- > Finally, is this even the fastest way to scroll through a large image?
- > That's the motivation behind using APP\_SCROLL in the first place. Should I
- > be using any other techniques as well, like DEVICE\_COPYING from pixmaps or
- > anything like that?

I haven't done this in a LONG time, but I used a Device Copy from a much larger pixmap into the smaller draw widget, using the values obtained from the VIEWPORT event structure to tell me where to start copying in my pixmap. It worked great, as far as I remember. I haven't been able to find that program, but I've changed computers several times since I wrote it.

Do you have a simple example you can provide us to fool around with?

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438, E-mail: [david@dfanning.com](mailto:david@dfanning.com)

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

---

Subject: Re: Expose Events (easy questions)  
Posted by [David Fanning](#) on Wed, 03 Apr 2002 17:35:39 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Ted Cary (tedcary@yahoo.com) writes:

> The APP\_SCROLL draw widget is in a resizable base

I have to admit, I don't get the APP\_SCROLL keyword!  
On Windows, at least, it appears to disable the  
scrolling ability of the operating system and \*makes\*  
you scroll the image yourself. (Tell me again, \*why\*  
do I want to do that when you already take care of it!?)

But, that said, the values in IDL 5.4 Windows of the  
X and Y fields of the event structure are the locations  
of the lower-left corner of the window in viewport  
coordinates, which is exactly what I would expect.

It is no problem copying the right part of the image  
from a pixmap, or from the image itself, using these  
values. But I still don't understand why you would  
want to. (I guess if you were going from the image  
data directly it would make some sense. But then it  
wouldn't be particularly fast.)

Cheers,

David

--

David W. Fanning, Ph.D.  
Fanning Software Consulting  
Phone: 970-221-0438, E-mail: [david@dfanning.com](mailto:david@dfanning.com)  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>  
Toll-Free IDL Book Orders: 1-888-461-0155

---

---

Subject: Re: Expose Events (easy questions)  
Posted by [Ted Cary](#) on Wed, 03 Apr 2002 17:49:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

"David Fanning" <[david@dfanning.com](mailto:david@dfanning.com)> wrote in message  
[news:MPG.1714eb51cd25064b989861@news.frii.com](mailto:news:MPG.1714eb51cd25064b989861@news.frii.com)...

> Ted Cary (tedcary@yahoo.com) writes:  
>  
>  
> But, that said, the values in IDL 5.4 Windows of the  
> X and Y fields of the event structure are the locations  
> of the lower-left corner of the window in viewport  
> coordinates, which is exactly what I would expect.

That's what I expected too, but at least on my Mac at work the X field does not appear to have anything to do viewport X-coordinates. It's the scr\_ysize, every time. I spent 3 hours toying with it yesterday, so I'm pretty sure about it. I'll post the code later.

I was hoping that by using APP\_SCROLL and then explicitly redrawing only the viewable portion of the window that I would speed things up. Drawing a larger image in an object graphics hierarchy to a scrollable window takes a long time--noticeably longer than drawing a smaller image. My cursor lags too much during drag-drawing with the mouse, when the hierarchy is redrawn with every motion event. The lag is definitely proportional to image size. The documentation for WIDGET\_DRAW maintains that one should always use APP\_SCROLL with large images, and in fact with anything drawn in device coordinates (I think). I was assuming this was because of the slowness problem? If not, what's the point?

---

---

Subject: Re: Expose Events (easy questions)  
Posted by [James Kuyper](#) on Wed, 03 Apr 2002 18:17:28 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning wrote:

> Ted Cary (tedcary@yahoo.com) writes:  
>  
>  
>> The APP\_SCROLL draw widget is in a resizable base  
>  
>  
> I have to admit, I don't get the APP\_SCROLL keyword!  
> On Windows, at least, it appears to disable the  
> scrolling ability of the operating system and \*makes\*  
> you scroll the image yourself. (Tell me again, \*why\*  
> do I want to do that when you already take care of it!?)  
>  
> But, that said, the values in IDL 5.4 Windows of the  
> X and Y fields of the event structure are the locations  
> of the lower-left corner of the window in viewport  
> coordinates, which is exactly what I would expect.

>  
> It is no problem copying the right part of the image  
> from a pixmap, or from the image itself, using these  
> values. But I still don't understand why you would  
> want to. (I guess if you were going from the image  
> data directly it would make some sense. But then it  
> wouldn't be particularly fast.)

I think the most reasonable use of this would be to implement something that uses the same interface as scrolling, but does something different from what the built-in does. For instance, a vertical scroll that takes you through the sections of a document at a constant rate, even though some sections are much larger than others. Or an image might be displayed with different magnifications, depending upon which part you're viewing.

---

Subject: Re: Expose Events (easy questions)  
Posted by [Ted Cary](#) on Thu, 04 Apr 2002 16:38:57 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning wrote:

> If you want my opinion, I'd stay as far away  
> from the APP\_SCROLL keyword as possible.

The behavior on our Macs here has nothing to do with APP\_SCROLL. I replaced the APP\_SCROLL switch with /VIEWPORT\_EVENTS, /EXPOSE\_EVENTS. No difference. I've tested the program on 3 different Macs now, even an iMac, and the X field of the expose event structure always contains the y screen size. Did anyone else test this on a Macintosh?

I'm still wondering why the documentation says to use this technique "when displaying images or anything else displayed in device units." It's supposed to be "good for displaying large images because the entire image does not have to be redrawn when viewport events are generated."

Is setting RETAIN to 0 and then handling expose and viewport motion events really the correct way to deal with large images? The documentation makes it sound like it's the preferred way, but nobody seems to have ever used it. I'll write something to test if this method is really any better... it's just hard to believe they'd make such claims in the documentation for no reason.

Thanks to everyone who took the time to test this out,

Ted Cary

---

---

Subject: Re: Expose Events (easy questions)

Posted by [David Fanning](#) on Thu, 04 Apr 2002 17:59:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Ted Cary (tedcary@yahoo.com) writes:

- > it's just hard to believe they'd make such claims
- > in the documentation for no reason.

I guess you have never worked for a software company.

I can think of 10 reasons just off the top of my head. :-)

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438, E-mail: [david@dfanning.com](mailto:david@dfanning.com)

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

---