

Craig Hamilton (someone@microsoft.com) writes:

- > In August I asked about a solution to the problem of "unable to create
- > pixmap" when running IDL on MSWin 2000. The Most Noble David Fanning explained
- > that pixmaps are created in video RAM, and, unless your video card is
- > "smart", I am limited by the video card. He also suggested a workaround using unmapped
- > draw widgets instead of pixmaps.
- >
- > I have questions about these two potential solutions:
- >
- > 1. Get a smarter video card (with loads of video RAM, I presume). Can
- > anyone suggest a video card that can use system RAM when necessary? I have
- > tried a GeForce4 4200 AGP card with 128 MB of RAM with no success trying
- > to allocate about 1500 pixmaps of total size of around 90 MB.
- > Interestingly, a Radeon VE with 32MB of video RAM works with roughly the same limits
- > as the GeForce4 card. So, it doesn't seem to be as simple as just throwing
- > more video RAM at the problem.
- >
- > 2. Use unmapped draw widgets instead of pixmaps. I started working on this
- > and ran into the problem (at least according to the documentation) that
- > mapping/unmapping applies only to base widgets, not draw widgets.
- >
- > So, if I unmapped a draw widget, it goes up to its parent base and unmaps
- > it, which is not what I want.

I don't know which video card is "smarter". Maybe Randy Frank is still listening in. He will know.

If I said "unmapped" draw widget, I'm sorry. At the time I was answering the question we were mapping and un mappingg "displays", which consisted of a base widget, a draw widget, and pixmap. We thought of them as "images", so I probably confused you by using imprecise language.

But we found that even the noble solution mentioned above didn't work so well in practice. :-(

After 30-40 images, we still found ourselves running out of window resources, and--of course--our client wanted to have *hundreds* of images open at once, after they got a look at our software and what it could do. :-)

We have since gone to what I call the "smoke and mirrors" approach to the problem. Fortunately for us, our design

makes it possible to only view one "image" at a time, although you can select any one of the hundreds of images in the stack. In practice, the user usually will select the "previous" or "next" image.

We reasoned that while the user was looking at the currently selected image, we could be doing some fancy footwork. So we designed our "pixmap" so that they actually create a pixmap window (and use window resources) only when they absolutely have to. Most of the time, they just carry around a pointer to an image that they *would* use as the pixmap, if they had to. Thus, the current, previous, and next images use pixmaps, but anything else has to create a pixmap window when requested.

This results in instantaneous display of the previous and next image, but there is a slight delay if the user suddenly wants to go to (for example) the first image in the stack. But the delay is not onerous (a momentary blink), and the up-side is that we can now load as many images into our system as required (limited only by the virtual memory available for paging).

Of course, all of this (displays, draw widgets, pixmaps, etc.) are wrapped up as objects so they are small, smart, and self-contained. They are quite easy to work with (well, once you get the hang of it). The beauty of the system is that we could completely re-work the way it all worked just by changing the code in a single object. If you have ever tried to do this in a non-object system, you can appreciate (again!) the power of object programming. :-)

Cheers,

David

--

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

Subject: Re: IDL/MSWin pixmap limitations, Part 2
Posted by [Craig Hamilton](#) on Mon, 25 Nov 2002 17:45:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

"David Fanning" <david@dfanning.com> wrote in message

news:MPG.184c036b8f953fec989a4b@news.frii.com...

> Craig Hamilton (someone@microsoft.com) writes:

>

>> In August I asked about a solution to the problem of "unable to create
>> pixmap" when running IDL on MSWin 2000. The Most Noble David Fanning
explained

>> that pixmaps are created in video RAM, and, unless your video card is
>> "smart", I am limited by the video card. He also suggested a workaround
using unmapped

>> draw widgets instead of pixmaps.

>>

>> I have questions about these two potential solutions:

>>

>> 1. Get a smarter video card (with loads of video RAM, I presume). Can
>> anyone suggest a video card that can use system RAM when necessary? I
have

>> tried a GeForce4 4200 AGP card with 128 MB of RAM with no success trying
>> to allocate about 1500 pixmaps of total size of around 90 MB.

>> Interestingly, a Radeon VE with 32MB of video RAM works with roughly the
same limits

>> as the GeForce4 card. So, it doesn't seem to be as simple as just
throwing

>> more video RAM at the problem.

>>

>> 2. Use unmapped draw widgets instead of pixmaps. I started working on
this

>> and ran into the problem (at least according to the documentation)
that

>> mapping/unmapping applies only to base widgets, not draw widgets.

>>

>> So, if I unmapped a draw widget, it goes up to its parent base and
unmaps

>> it, which is not what I want.

>

> I don't know which video card is "smarter". Maybe Randy Frank
> is still listening in. He will know.

>

> If I said "unmapped" draw widget, I'm sorry. At the time I
> was answering the question we were mapping and un mappingg
> "displays", which consisted of a base widget, a draw widget,
> and pixmap. We thought of them as "images", so I probably
> confused you by using imprecise language.

>

> But we found that even the noble solution mentioned above
> didn't work so well in practice. :-(

>

> After 30-40 images, we still found ourselves running out
> of window resources, and--of course--our client wanted to

> have *hundreds* of images open at once, after they got a look
> at our software and what it could do. :-)
>
> We have since gone to what I call the "smoke and mirrors"
> approach to the problem. Fortunately for us, our design
> makes it possible to only view one "image" at a time,
> although you can select any one of the hundreds of images
> in the stack. In practice, the user usually will select
> the "previous" or "next" image.
>
> We reasoned that while the user was looking at the currently
> selected image, we could be doing some fancy footwork. So we
> designed our "pixmap" so that they actually create a pixmap
> window (and use window resources) only when they absolutely
> have to. Most of the time, they just carry around a pointer
> to an image that they *would* use as the pixmap, if they
> had to. Thus, the current, previous, and next images use
> pixmaps, but anything else has to create a pixmap window
> when requested.
>
> This results in instantaneous display of the previous and
> next image, but there is a slight delay if the user suddenly
> wants to go to (for example) the first image in the stack.
> But the delay is not onerous (a momentary blink), and the
> up-side is that we can now load as many images into our system
> as required (limited only by the virtual memory available for
> paging).
>
> Of course, all of this (displays, draw widgets, pixmaps, etc.)
> are wrapped up as objects so they are small, smart, and self-contained.
> They are quite easy to work with (well, once you get the hang of it).
> The beauty of the system is that we could completely re-work the
> way it all worked just by changing the code in a single object.
> If you have ever tried to do this in a non-object system, you
> can appreciate (again!) the power of object programming. :-)
>
> Cheers,
>
> David
>
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
> Toll-Free IDL Book Orders: 1-888-461-0155

Wowzers, David, that is the speediest newsgroup response I have ever

gotten.... thanks.

So, it looks like the software solution you mention below wouldn't work for switching rapidly between animations, right? I want an instantaneous change from a 30 frame cinelooop (1024x512) to one of 15 other ones the same size....

(This works great on Unix.)

Perhaps I can track down a "smart" video card.

Thanks so much for your help,
Craig

(I heard about this great book on coyotes that I am finally going to order....

something about coyote programmers)

Subject: Re: IDL/MSWin pixmap limitations, Part 2
Posted by [Rick Towler](#) on Mon, 25 Nov 2002 17:55:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Craig Hamilton" <someone@microsoft.com> wrote

- > In August I asked about a solution to the problem of "unable to
- > create pixmap" when running IDL on MSWin 2000. The Most Noble
- > David Fanning explained that pixmaps are created in video RAM,
- > and, unless your video card is "smart", I am limited by the
- > video card. He also suggested a workaround using unmapped draw
- > widgets instead of pixmaps.
- >
- > I have questions about these two potential solutions:
- >
- > 1. Get a smarter video card (with loads of video RAM, I presume).
- > Can anyone suggest a video card that can use system RAM when
- > necessary?

Most all video cards on the PC can use system RAM if necessary. My guess is that this is not a RAM issue, but a driver issue. Your experiences with the 128MB nVidia card and the 32MB ATI card although not definitive support this. The drivers probably set some limit regardless of the system resources available.

But, just for fun try upping your "AGP Aperture Size" in your BIOS:

"This option selects the size of the AGP aperture. The aperture is a portion of the PCI memory address range dedicated as graphics memory address space. Host cycles that hit the aperture range are forwarded to the AGP without

need for translation. This size also determines the maximum amount of system RAM that can be allocated to the graphics card for texture storage."

Note that the actual amount of ram available is something like (AGP aperture size - 12) / 2.

I still think your driver is limiting but give this a shot. The standard disclaimers apply.

-Rick

Subject: Re: IDL/MSWin pixmap limitations, Part 2
Posted by [Pavel A. Romashkin](#) on Mon, 25 Nov 2002 19:04:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

I thought that if you needed a large number of images in a pixmap (which would still fit in VRAM) but ran out of windowing resources you could make one very big pixmap window and load it with sequences of frames, then just display those frames, choosing their coordinates programmatically. One of your programs did that, I recall. Of course some flexibility is lost but you could also "repaint" the pixmap in the background and have more than just 3 frames ready for instant displaying.

Cheers,
Pavel

David Fanning wrote:

>
> Craig Hamilton (someone@microsoft.com) writes:
>
>
> I don't know which video card is "smarter". Maybe Randy Frank
> is still listening in. He will know.
>
> If I said "unmapped" draw widget, I'm sorry. At the time I
> was answering the question we were mapping and un mappingg
> "displays", which consisted of a base widget, a draw widget,
> and pixmap. We thought of them as "images", so I probably
> confused you by using imprecise language.
>
> But we found that even the noble solution mentioned above
> didn't work so well in practice. :-(
>
> After 30-40 images, we still found ourselves running out
> of window resources, and--of course--our client wanted to
> have *hundreds* of images open at once, after they got a look

> at our software and what it could do. :-)
>
> We have since gone to what I call the "smoke and mirrors"
> approach to the problem. Fortunately for us, our design
> makes it possible to only view one "image" at a time,
> although you can select any one of the hundreds of images
> in the stack. In practice, the user usually will select
> the "previous" or "next" image.
>
> We reasoned that while the user was looking at the currently
> selected image, we could be doing some fancy footwork. So we
> designed our "pixmap" so that they actually create a pixmap
> window (and use window resources) only when they absolutely
> have to. Most of the time, they just carry around a pointer
> to an image that they *would* use as the pixmap, if they
> had to. Thus, the current, previous, and next images use
> pixmaps, but anything else has to create a pixmap window
> when requested.
>
> This results in instantaneous display of the previous and
> next image, but there is a slight delay if the user suddenly
> wants to go to (for example) the first image in the stack.
> But the delay is not onerous (a momentary blink), and the
> up-side is that we can now load as many images into our system
> as required (limited only by the virtual memory available for
> paging).
>
> Of course, all of this (displays, draw widgets, pixmaps, etc.)
> are wrapped up as objects so they are small, smart, and self-contained.
> They are quite easy to work with (well, once you get the hang of it).
> The beauty of the system is that we could completely re-work the
> way it all worked just by changing the code in a single object.
> If you have ever tried to do this in a non-object system, you
> can appreciate (again!) the power of object programming. :-)
>
> Cheers,
>
> David
>
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
> Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: IDL/MSWin pixmap limitations, Part 2
Posted by [David Fanning](#) on Mon, 25 Nov 2002 19:30:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Craig Hamilton (someone@microsoft.com) writes:

> So, it looks like the software solution you mention below wouldn't work for
> switching rapidly between animations, right? I want an instantaneous change
> from a 30 frame cinelooop (1024x512) to one of 15 other ones the same
> size....

Depending upon how fast we want animations to be, we can switch between what we call "retained" pixmaps (i.e., real pixmaps) and "non-retained" pixmaps (those that have no real window associated with them). Most of the time, with retained pixmaps, we have to turn the animation speed down because the animation occurs too rapidly. We find that for the satellite images we are working with that animations in non-retained mode are perfectly adequate for our purposes, so this has been a good solution for us.

In any case, the system is flexible enough that we can turn retained and non-retained pixmaps on and off at will, so we can monitor window resources (after a fashion) and alert the user if the animation size they have requested (for example) is too large for retained pixmaps. They have the option then of reducing the size of the animation (either window size or number of frames), or using non-retained pixmaps.

Users who are used to PCs understand there are trade-offs. :-)

Cheers,

David

--

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