

Forum:[Hack Forums](#)

Thread: [How does Imminent Monitor provides such high speed for Remote Desktop?](#)

jim_377 - March 31, 2016, 11:55 am

Imminent Monitor has a very high speed for Remote Desktop (30+ FPS).

Does anyone knows what approach it uses to achieve such high speed?

webdev77 - March 31, 2016, 11:58 am

Instead of looking at just one program, try looking at how live streaming works and the tech behind it. If you google live streaming programs, you will find some relevant results.

\$lick - March 31, 2016, 2:45 pm

Imminent Monitor uses the same method as NanoCore as far as I'm aware, it only sends over the information that has changed and not the entire screen.

nuk - April 1, 2016, 11:39 pm

Why not just check out the source to Imminent Monitor? Oh... here ya go ;)

Imminent Monitor Source: <http://pastebin.com/bsfmNp4c>

Enjoy... let me know how you make out ;)

Odus - April 2, 2016, 2:25 am

(Apr 1, 2016 11:39 PM)nuk Wrote: [\[CITE\]](#) Why not just check out the source to Imminent Monitor? Oh... here ya go ;)

Imminent Monitor Source: <http://pastebin.com/bsfmNp4c>

Enjoy... let me know how you make out ;)

I'd hope they've made some kind of improvements to it after 3 years though. ?

.Monarch. - April 2, 2016, 8:17 am

(Mar 31, 2016 11:55 AM)jim_377 Wrote: [\[CITE\]](#) Imminent Monitor has a very high speed for Remote Desktop (30+ FPS).

Does anyone knows what approach it uses to achieve such high speed?

I don't own Imminent Monitor but this is what they might be doing.

- Custom compression algo for images
- Motion Detection (This way they only send the data which changes) this saves lot's of bandwidth.

Ninja - April 2, 2016, 8:28 am

(Apr 1, 2016 11:39 PM)nuk Wrote: [\[CITE\]](#) Why not just check out the source to Imminent Monitor? Oh... here ya go ;)

Imminent Monitor Source: <http://pastebin.com/bsfmNp4c>

Enjoy... let me know how you make out ;)

That source code gave me cancer.

Adderall - April 2, 2016, 8:38 am

(Apr 2, 2016 08:17 AM).Zeref. Wrote: [\[CITE\]](#) I don't own Imminent Monitor but this is what they might be doing.

- Custom compression algo for images
- Motion Detection (This way they only send the data which changes) this saves lot's of bandwidth. Use JPEG compression and write the image to bytes using a stream and send it to the server and let it decompress it to an image.

(Apr 2, 2016 08:28 AM)Ninja Wrote: [\[CITE\]](#) That source code gave me cancer. My computer died of clicking on the source. - Google chrome has stopped working...

Zilog80 - April 2, 2016, 2:35 pm

(Apr 2, 2016 08:28 AM)Ninja Wrote: [\[CITE\]](#) That source code gave me cancer. I thought I was the only one who thought so?

"On Error Resume Next" gave me cancer+ebola+Aids+Lupus

For the @OP, you should take a look for "Desktop Duplication API". This API uses DirectX 11 for very fast imagen stuff, like:

IDXGIOutputDuplication::GetFrameDirtyRects

"returns dirty regions, which are non-overlapping rectangles that indicate the areas of the desktop image that the operating system updated since you processed the previous desktop image."

IDXGIOutputDuplication::GetFrameMoveRects

"returns move regions, which are rectangles of pixels in the desktop image that the operating system moved to another location within the same image. Each move region consists of a destination rectangle and a source point. The source point specifies the location from where the operating system copied the region and the destination rectangle specifies to where the operating system moved that region. Move regions are always non-stretched regions so the source is always the same size as the destination."

You can use SharpDX in order to use this methods in C# or VB.NET

Combining SharpDX with any kind of compression algorithim I made my own "remote screener streamer" (private use, not public) capable to streamings to 1920 x 1080 24bpp @ 0,56 mbps on high spikes (notice mbps are mega bits, not mega Bytes). Performance (static & changing images) ups to 60 FPS in ADSL with 512 Kb upload Channel. (Very low quality ADSL, I know)

jim_377 - April 2, 2016, 7:05 pm

(Apr 2, 2016 02:35 PM)Zilog80 Wrote: [\[CITE\]](#) I thought I was the only one who thought so?

"On Error Resume Next" gave me cancer+ebola+Aids+Lupus

For the @OP, you should take a look for "Desktop Duplication API". This API uses DirectX 11 for very fast imagen stuff, like:

IDXGIOutputDuplication::GetFrameDirtyRects

"returns dirty regions, which are non-overlapping rectangles that indicate the areas of the desktop image that the operating system updated since you processed the previous desktop image."

IDXGIOutputDuplication::GetFrameMoveRects

"returns move regions, which are rectangles of pixels in the desktop image that the operating system moved to another location within the same image. Each move region consists of a

destination rectangle and a source point. The source point specifies the location from where the operating system copied the region and the destination rectangle specifies to where the operating system moved that region. Move regions are always non-stretched regions so the source is always the same size as the destination."

You can use SharpDX in order to use this methods in C# or VB.NET

Combining SharpDX with any kind of compression algorithm I made my own "remote screener streamer" (private use, not public) capable to streamings to 1920 x 1080 24bpp @ 0,56 mbps on high spikes (notice mbps are mega bits, not mega Bytes). Performance (static & changing images) ups to 60 FPS in ADSL with 512 Kb upload Channel. (Very low quality ADSL, I know)

Unfortunately the Desktop Duplication API only works for Windows 8 and later ?.

Zilog80 - April 3, 2016, 3:10 am

(Apr 2, 2016 07:05 PM)jim_377 Wrote: [\[CITE\]](#) Unfortunately the Desktop Duplication API only works for Windows 8 and later ?.

Not really with this Windows 7 Update:

[msdn.microsoft.com/en-us/library/windows/desktop/jj863687\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/jj863687(v=vs.85).aspx)

This Update provides Win7 ti use DXGI 1.2 API.

ronald51 - April 3, 2016, 8:48 am

Thanks nuk for the source code.

jim_377 - April 3, 2016, 9:17 am

(Apr 3, 2016 03:10 AM)Zilog80 Wrote: [\[CITE\]](#) Not really with this Windows 7 Update:

[msdn.microsoft.com/en-us/library/windows/desktop/jj863687\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/jj863687(v=vs.85).aspx)

This Update provides Win7 ti use DXGI 1.2 API.

Great news. Thanks man for pointing that out.

Is there any drawbacks to the Desktop Duplication API, I read something about that some windows can choose not be captured by this API:

blogs.msdn.microsoft.com/dsui_team/2013/03/25/ways-to-capture-the-screen/

Is this true or I misunderstood what I've read ??

Slavoj Žižek - April 3, 2016, 10:07 am

it only sends the parts of the screen that have changed, and it uses hardware acceleration for the compression (I think)

Zilog80 - April 3, 2016, 10:17 am

(Apr 3, 2016 09:17 AM)jim_377 Wrote: [\[CITE\]](#) Great news. Thanks man for pointing that out.

Is there any drawbacks to the Desktop Duplication API, I read something about that some windows can choose not be captured by this API:

blogs.msdn.microsoft.com/dsui_team/2013/03/25/ways-to-capture-the-screen/

Is this true or I misunderstood what I've read ??

As you can read on your provided link, there is a drawback related with Fullscreen apps.

Fullscreen DirectX (and OpenGL) applications can not be captured without source code control. If the SwapChain hasn't been created with the right flags, there is (at least without any system-hacking-skills) no way to duplicate the screen. You can find more information about that on the SwapChain Flags MSDN page:

[msdn.microsoft.com/de-de/library/windows/desktop/bb173076\(v=vs.85\).aspx](https://msdn.microsoft.com/de-de/library/windows/desktop/bb173076(v=vs.85).aspx)

As you can see it you will either have to use `IDXGIFactory2::CreateSwapChainForHwnd`, `IDXGIFactory2::CreateSwapChainForCoreWindow` or `IDXGIFactory2::CreateSwapChainForComposition` without the `DXGI_SWAP_CHAIN_FLAG_DISPLAY_ONLY` to allow other applications to access the SwapChain's backbuffer.

Anyway, even this drawback, Desktop Duplication API (GXDI 1.2) smash classic & old school BitBlt GDI+ capture screen.

jim_377 - April 3, 2016, 2:43 pm

(Apr 3, 2016 10:17 AM)Zilog80 Wrote: [\[CITE\]](#) As you can read on your provided link, there is a drawback related with Fullscreen apps.

Fullscreen DirectX (and OpenGL) applications can not be captured without source code control. If the SwapChain hasn't been created with the right flags, there is (at least without any system-hacking-skills) no way to duplicate the screen. You can find more information about that on the SwapChain Flags MSDN page:

[msdn.microsoft.com/de-de/library/windows/desktop/bb173076\(v=vs.85\).aspx](https://msdn.microsoft.com/de-de/library/windows/desktop/bb173076(v=vs.85).aspx)

As you can see it you will either have to use `IDXGIFactory2::CreateSwapChainForHwnd`, `IDXGIFactory2::CreateSwapChainForCoreWindow` or `IDXGIFactory2::CreateSwapChainForComposition` without the `DXGI_SWAP_CHAIN_FLAG_DISPLAY_ONLY` to allow other applications to access the SwapChain's backbuffer.

Anyway, even this drawback, Desktop Duplication API (GXDI 1.2) smash classic & old school BitBlt GDI+ capture screen.

Thanks a lot for your help, I just have one last question, does the mouse cursor blink when capturing the screen using the Desktop Duplication API (because in GDI it does blink)?