

Forum:[Hack Forums](#)

Thread: [ScreenCapture \[Faster Screenshots\]](#)

Malum™ - May 7, 2016, 4:53 pm

I needed a faster way to take screenshots so I threw this class together because Graphics.CopyFromScreen was too slow for what I was working on.

<3 BitBlt

VB.NET

Spoiler ([Click to View](#))

Code NotInheritable Class ScreenCapture

Private Sub New()

End Sub

#Region "Native API"

<DllImport("gdi32.dll", EntryPoint := "BitBlt", SetLastError := True)> _

Private Shared Function BitBlt(<[In]> hdc As IntPtr, nXDest As Integer, nYDest As Integer, nWidth As Integer, nHeight As Integer, <[In]> hdcSrc As IntPtr, _
nXSrc As Integer, nYSrc As Integer, dwRop As Integer) As

<MarshalAs(UnmanagedType.Bool)> Boolean

End Function

<DllImport("gdi32.dll")> _

Private Shared Function CreateDC(lpszDriver As String, lpszDevice As String, lpszOutput As String, lpInitData As IntPtr) As IntPtr

End Function

<DllImport("gdi32.dll")> _

Private Shared Function DeleteDC(<[In]> hdc As IntPtr) As Boolean

End Function

#End Region

Public Shared Function GetScreenshot() As Bitmap

Dim screen__1 As New Bitmap(Screen.PrimaryScreen.Bounds.Width, Screen.PrimaryScreen.Bounds.Height, PixelFormat.Format32bppArgb)

Return GetFromScreen(New Point(0, 0), screen__1.Size)

End Function

Public Shared Function GetFromScreen(start As Point, size As Size) As Bitmap

Dim screen As New Bitmap(size.Width, size.Height, PixelFormat.Format32bppArgb)

Using g As Graphics = Graphics.FromImage(screen)

Dim destDeviceContext As IntPtr = g.GetHdc()

Dim srcDeviceContext As IntPtr = CreateDC("DISPLAY", Nothing, Nothing, IntPtr.Zero)

BitBlt(destDeviceContext, 0, 0, screen.Width, screen.Height, srcDeviceContext, _
start.X, start.Y, &Hcc0020)

DeleteDC(srcDeviceContext)

```
g.ReleaseHdc(destDeviceContext)
End Using
```

```
Return screen
End Function
End Class
```

C#

Spoiler ([Click to View](#))

Code static class ScreenCapture

```
{
    #region Native API
    [DllImport("gdi32.dll", EntryPoint = "BitBlt", SetLastError = true)]
    [return: MarshalAs(UnmanagedType.Bool)]
    private static extern bool BitBlt([In] IntPtr hdc, int nXDest, int nYDest, int nWidth, int nHeight,
    [In] IntPtr hdcSrc, int nXSrc, int nYSrc, int dwRop);

    [DllImport("gdi32.dll")]
    private static extern IntPtr CreateDC(string lpszDriver, string lpszDevice, string lpszOutput,
    IntPtr lpInitData);

    [DllImport("gdi32.dll")]
    private static extern bool DeleteDC([In] IntPtr hdc);
    #endregion

    public static Bitmap GetScreenshot()
    {
        Bitmap screen = new Bitmap(Screen.PrimaryScreen.Bounds.Width,
        Screen.PrimaryScreen.Bounds.Height, PixelFormat.Format32bppArgb);
        return GetFromScreen(new Point(0, 0), screen.Size);
    }

    public static Bitmap GetFromScreen(Point start, Size size)
    {
        Bitmap screen = new Bitmap(size.Width, size.Height, PixelFormat.Format32bppArgb);

        using (Graphics g = Graphics.FromImage(screen))
        {
            IntPtr destDeviceContext = g.GetHdc();
            IntPtr srcDeviceContext = CreateDC("DISPLAY", null, null, IntPtr.Zero);

            BitBlt(destDeviceContext, 0, 0, screen.Width, screen.Height, srcDeviceContext, start.X,
            start.Y, 0x00CC0020);

            DeleteDC(srcDeviceContext);
            g.ReleaseHdc(destDeviceContext);
        }

        return screen;
    }
}
```

Usage:

Code //C#

```
Bitmap screen = ScreenCapture.GetScreenshot()
```

```
Bitmap partofscreen = Screen.GetFromScreen(new Point(530, 319), 100, 100);
```

'VB.NET

```
Dim screen As New Bitmap = ScreenCapture.GetScreenshot()
```

```
Dim partofscreen As New Bitmap = Screen.GetFromScreen(new Point(530, 319), 100, 100)
```

6e756c6c - May 7, 2016, 11:41 pm

I'm not 100% sure without testing because I handle this a bit differently, but you may run into scaling issues with this code if the computer's DPI isn't set to 100%. You may want to test for that! :)

GuRuHaX - May 8, 2016, 2:37 am

Thanks a lot for the sharing, appreciate it ! Will add this to my library collection ;)

PrivateDonut - May 8, 2016, 2:39 am

Looks pretty good, I have always wanted a Screen Capture program that would allow me to upload directly to a FTP server, so I would have better control over all my images.

Malum™ - May 8, 2016, 8:18 am

(May 7, 2016 11:41 PM)Aeonhack Wrote: [\[CITE\]](#) I'm not 100% sure without testing because I handle this a bit differently, but you may run into scaling issues with this code if the computer's DPI isn't set to 100%. You may want to test for that! :)

Didn't even occur to me, I'm on mobile right now and I'll definitely do some testing later. Thanks for the suggestion!

Adderall - May 9, 2016, 9:28 am

(May 7, 2016 11:41 PM)Aeonhack Wrote: [\[CITE\]](#) I'm not 100% sure without testing because I handle this a bit differently, but you may run into scaling issues with this code if the computer's DPI isn't set to 100%. You may want to test for that! :)

I tested it and it will not show the entire screen if the DPI is not 100%.

Spoiler [\(Click to View\)](#)

I don't know how to fix it apparently G.CopyFromScreen has GDI issues too. Maybe someone else knows how to fix the issue.

Malum™ - May 9, 2016, 1:04 pm

I found a fix for what Aeonhack had suggested which was to use Graphics.DpiX / Graphics.DpiY.

Lexus™ - May 10, 2016, 6:20 am

thanks for sharing this with us ;)

Luciffer - May 13, 2016, 2:28 am

try this script

Code 'for timers

```
Dim bounds As Rectangle
```

```
Dim screenshot As System.Drawing.Bitmap
```

```
Dim graph As Graphics
```

```
bounds = Screen.PrimaryScreen.Bounds
```

```
screenshot = New System.Drawing.Bitmap(bounds.Width, bounds.Height,
```

```
System.Drawing.Imaging.PixelFormat.Format32bppArgb)
    graph = Graphics.FromImage(screenshot)
    graph.CopyFromScreen(bounds.X, bounds.Y, 0, 0, bounds.Size,
CopyPixelOperation.SourceCopy)
    PictureBox1.Image = screenshot
    Timer1.Enabled = False
    Me.Opacity = 100
```

'For button when captured

```
PictureBox1.Show()
Me.Opacity = 0
Timer1.Enabled = True
Button2.Enabled = True
```

For save the file

```
Dim savefiledialog1 As New SaveFileDialog
Try
    savefiledialog1.Title = "Save File"
    savefiledialog1.FileName = "Capture"
    savefiledialog1.Filter = "Bitmap (*.bmp;*.dib)|*.bmp;*.dib|JPEG
(*.jpg;*.jpeg;*.jpe;*.jfif)|*.jpg;*.jpeg;*.jpe;*.jfif|GIF (*.gif)|*.gif|TIFF (*.tif;*.tiff)|*.tif;*.tiff|PNG
(*.png)|*.png"
    If savefiledialog1.ShowDialog() = DialogResult.OK Then
        PictureBox1.Image.Save(savefiledialog1.FileName,
System.Drawing.Imaging.ImageFormat.Bmp)
    End If
Catch ex As Exception
    ' Do Nothing
End Try
```

Ninja - May 13, 2016, 5:18 am

(May 13, 2016 02:28 AM)Luciffer Wrote: [CITE](#) try this script

Code 'for timers

```
Dim bounds As Rectangle
Dim screenshot As System.Drawing.Bitmap
Dim graph As Graphics
bounds = Screen.PrimaryScreen.Bounds
screenshot = New System.Drawing.Bitmap(bounds.Width, bounds.Height,
System.Drawing.Imaging.PixelFormat.Format32bppArgb)
graph = Graphics.FromImage(screenshot)
graph.CopyFromScreen(bounds.X, bounds.Y, 0, 0, bounds.Size,
CopyPixelOperation.SourceCopy)
PictureBox1.Image = screenshot
Timer1.Enabled = False
Me.Opacity = 100
```

'For button when captured

```
PictureBox1.Show()  
Me.Opacity = 0  
Timer1.Enabled = True  
Button2.Enabled = True
```

For save the file

```
Dim savefiledialog1 As New SaveFileDialog  
Try  
    savefiledialog1.Title = "Save File"  
    savefiledialog1.FileName = "Capture"  
    savefiledialog1.Filter = "Bitmap (*.bmp;*.dib)|*.bmp;*.dib|JPEG  
(* .jpg;*.jpeg;*.jpe;*.jfif)|*.jpg;*.jpeg;*.jpe;*.jfif|GIF (*.gif)|*.gif|TIFF (*.tif;*.tiff)|*.tif;*.tiff|PNG  
(* .png)|*.png"  
    If savefiledialog1.ShowDialog() = DialogResult.OK Then  
        PictureBox1.Image.Save(savefiledialog1.FileName,  
System.Drawing.Imaging.ImageFormat.Bmp)  
    End If  
Catch ex As Exception  
    ' Do Nothing  
End TryAre you retarded..?
```

Zilog80 - May 13, 2016, 6:53 am

Just testing "how much faster"

Malem VS Zilog80 Screenshot ([Click to View](#))

Edit:

The above results: Windows 10. C2Duo @ 2.67 Ghz. 4 GB RAM.\$25 cheap NVIDIA graphics card. 1280 x 1024

The next results: Windows 10. i7 4790K @ 4.0 Ghz. 16 GB RAM. 970 NVIDIA graphics card. 1920 x 1080: ([Click to View](#))

WTF ?

ya3ko - May 13, 2016, 4:02 pm

thanks for sharing this with us ;)

Malum™ - May 13, 2016, 6:39 pm

(May 13, 2016 06:53 AM)Zilog80 Wrote: ([CITE](#)) Just testing "how much faster"

Malem VS Zilog80 Screenshot ([Click to View](#))

Edit:

The above results: Windows 10. C2Duo @ 2.67 Ghz. 4 GB RAM.\$25 cheap NVIDIA graphics card. 1280 x 1024

The next results: Windows 10. i7 4790K @ 4.0 Ghz. 16 GB RAM. 970 NVIDIA graphics card. 1920 x 1080: [\(Click to View\)](#)

WTF ?

I'm not using DirectX, of course it's not going to come close to that. Lol.

He B TeMy - May 14, 2016, 1:37 pm

You can't get below 50ms constantly in 1080p with WINAPI, so, you'll get at most 20 fps, if i recall, the tops were really 15-16, this is with Aero ON, which is default since Vista days.

I don't know how exactly Nanocore works but i'm pretty sure it's not 30+ full screenshots per second, he probably handles it on a changed area of sorts, it works quite good but don't expect to hit 30 without DirectX, and nanocore doesn't use DirectX, nor do i know if it's possible or that easy.

Ninja - May 14, 2016, 3:30 pm

(May 14, 2016 01:37 PM)ff20 Wrote: [\[CITE\]](#) You can't get below 50ms constantly in 1080p with WINAPI, so, you'll get at most 20 fps, if i recall, the tops were really 15-16, this is with Aero ON, which is default since Vista days.

I don't know how exactly Nanocore works but i'm pretty sure it's not 30+ full screenshots per second, he probably handles it on a changed area of sorts, it works quite good but don't expect to hit 30 without DirectX, and nanocore doesn't use DirectX, nor do i know if it's possible or that easy.

It checks what has changed on the screen and sends the changes. Pretty sure it still captures the whole screen, but for bandwidth reasons it only sends the changes. At least this is what I heard.

HFDone - May 14, 2016, 3:53 pm

no reason for remote control to use directx we not going to watch movies

Zilog80 - May 15, 2016, 3:39 am

More testing (memory usage for 60 frames captured)

Malem VS Zilog80 Screenshot [\(Click to View\)](#)

He B TeMy - May 15, 2016, 2:20 pm

(May 14, 2016 03:30 PM)Ninja Wrote: [\[CITE\]](#) It checks what has changed on the screen and sends the changes. Pretty sure it still captures the whole screen, but for bandwidth reasons it only sends the changes. At least this is what I heard.

I haven't tested nanocore on 1080p but i bet it doesn't do that, the fps are just too high to do that.

I'd love to test it, but i can't atm, if Aeonhack would be kind to confirm this...

Adderall - May 15, 2016, 3:48 pm

(May 15, 2016 02:20 PM)ff20 Wrote: [\[CITE\]](#) I'd love to test it, but i can't atm, if Aeonhack would be kind to confirm this...

I think Nexus should confirm that, because he is the owner of NanoCore now, but he confirmed it once that the client only sends the changed part of the screen to the server. Most RATs use tha, so it isn't a very unique feature.

He B TeMy - May 16, 2016, 3:59 am

(May 15, 2016 03:48 PM)Timsel10 Wrote: [\[CITE\]](#) I think Nexus should confirm that, because he is

the owner of NanoCore now, but he confirmed it once that the client only sends the changed part of the screen to the server. Most RATs use tha, so it isn't a very unique feature.

But other RAT's don't have 40 fps at somepoint in rdp, AFAIK the WINAPI doesn't permit downscale when you take a screenshot, so, even if the window in the client is not the highest resolution possible you can only take the screenshot with the highest resolution and then downscale (which seems impossible to do 40fps), like i said, i want to test it myself but i can't now...

I know Aeonhack isn't the owner, but he knows better, no offense.

3mani_killer - May 16, 2016, 6:10 am

thanks for sharing this with us ;)

Malum™ - May 16, 2016, 10:11 am

I'm glad this huge discussion broke out between you guys. It's really interesting. I really appreciate the info.