

RE: Managed CameraCaptureDialog class causes an Unknown Exception.

Source:

<http://www.tech-archive.net/Archive/PocketPC/microsoft.public.pocketpc.developer/2007-01/msg00415.html>

- *From:* Noz <Noz@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 22 Jan 2007 12:36:04 -0800
-

Hi Tracy,

I've avoided using the managed classes and so have not come up with the solution to fixing this problem directly. I ended up working round this and writing a dll in embedded C++ that makes the same API call in unmanaged code. This works with no problems and allows me to get closer to the API itself. I can track errors better and respond to them more effectively. When writing this code I discovered a couple of little gocha's with the call. One is that the call fails if the file it tries to create already exists, in managed code the exception raised is not very graceful. There is no overloaded function in the API itself that you can call to overwrite the file if it exists. If this is a major support issue between the managed API and core API or/and issues with the WM devices then MS and its partners need to address this.

Even though this API gets results I'm not 100% impressed with it. The lack of control over the dialog, speed of the call and the issues surrounding the call do need improvement. I maybe a bit hypercritical here as they have made a giant leap here but it could be better <bg>. I'm looking at resorting to using the DirectShow API in the long term to control a mobile camera generically instead. This API is very complex but I think I found some good sample code to get me started that makes use of this API in a unmanaged/managed code relationship. Although the SDK samples have a DirectShow sample it doesn't give you enough to get into modifying it. The SDK docs for this API just suck! This API does give you more control on the performance and output. Although the sample I found is good the author wants to improve it by building a custom filtergraph to get more control on camera (media) input.

The code comes from a section of the MS mobile team who found a unique way to develop windows mobile. Its an open source project that allows you to turn a smartphone into a robot. It makes use of a taylored version of the WM DirectShow SDK Sample to capture still and motion images. As I said the author wants to improve image capture performance by building his own filter. I'm hoping I can sus this out myself at some point. Check the site out:

<http://www.wimobot.com/default.aspx>

RE: Managed CameraCaptureDialog class causes an Unknown Exception.

Hope this helps, good luck

Noz

"Tracy-LeeR" wrote:

I am getting exactly the same error, though it seems to occur at random intervals.

I have tried on the imate k-jam, jamin and the new jasjam.

I have tried everything from calling the garbage collector, disposing of objects, checking memory and background running processes etc.

The k-jam's and jamin's error occurs every now and then, while the jasjam on every second attempt to call the "CameraCaptureDialog.ShowDialog()". If I exit out of the application and run it again, it will capture the first image fine.

I have browsed the msdn forums and there are a number of developers with the same issue, though no one has been able to provide a solution. We have a number of users out in the field with devices running the application, so exiting and restarting everytime the user wants to take another photo is not a solution I can use.

Error Information is as follows:

HRESULT: -2146233079

InnerException: NULL

Message: An unknown error occurred

Stack Trace:

at

Microsoft.WindowsMobile.Forms.CameraCaptureDialog.LaunchCameraCaptureDialog()
at Microsoft.WindowsMobile.Forms.CameraCaptureDialog.ShowDialog().....

Any help, suggestions or input (or even better – the actual solution) would be greatly appreciated.

"Noz" wrote:

Hi All,

I tried to run the following code:

```
Microsoft.WindowsMobile.Forms.CameraCaptureDialog myPictureTaker =  
new Microsoft.WindowsMobile.Forms.CameraCaptureDialog();  
if (myPictureTaker.ShowDialog() == DialogResult.OK)  
{
```

RE: Managed CameraCaptureDialog class causes an Unknown Exception.

RE: Managed CameraCaptureDialog class causes an Unknown Exception.

```
string myPicture = myPictureTaker.FileName;

Image MyImage = (Image)new Bitmap(myPicture);

pictureBox1.Image = MyImage;

}

myPictureTaker.Dispose();
```

The line that shows the dialog 'myPictureTaker.ShowDialog()' raises an unknown exception when run on my Windows Mobile 5 IPaq hw6915. The actual exception details are:

```
System.InvalidOperationException was unhandled
Message="An unknown error occurred."
StackTrace:
at
Microsoft.WindowsMobile.Forms.CameraCaptureDialog.LaunchCameraCaptureDialog()
at Microsoft.WindowsMobile.Forms.CameraCaptureDialog.ShowDialog()
at StatusMWS.frmFinal.btnSave_Click()
at System.Windows.Forms.Control.OnClick()
at System.Windows.Forms.Button.OnClick()
at System.Windows.Forms.ButtonBase.WnProc()
at System.Windows.Forms.Control._InternalWnProc()
at Microsoft.AGL.Forms.EVL.EnterModalDialog()
at System.Windows.Forms.Form.ShowDialog()
at StatusMWS.frmReason.btnAccept_Click()
at System.Windows.Forms.Control.OnClick()
at System.Windows.Forms.Button.OnClick()
at System.Windows.Forms.ButtonBase.WnProc()
at System.Windows.Forms.Control._InternalWnProc()
at Microsoft.AGL.Forms.EVL.EnterModalDialog()
at System.Windows.Forms.Form.ShowDialog()
at StatusMWS.frmAddr.btnSelect_Click()
at System.Windows.Forms.Control.OnClick()
at System.Windows.Forms.Button.OnClick()
at System.Windows.Forms.ButtonBase.WnProc()
at System.Windows.Forms.Control._InternalWnProc()
at Microsoft.AGL.Forms.EVL.EnterModalDialog()
at System.Windows.Forms.Form.ShowDialog()
at StatusMWS.frmMain.btnMBReport_Click()
at System.Windows.Forms.Control.OnClick()
at System.Windows.Forms.Button.OnClick()
at System.Windows.Forms.ButtonBase.WnProc()
at System.Windows.Forms.Control._InternalWnProc()
at Microsoft.AGL.Forms.EVL.EnterMainLoop()
at System.Windows.Forms.Application.Run()
at StatusMWS.Program.Main()
```

RE: Managed CameraCaptureDialog class causes an Unknown Exception.

I try to run this code in a button click event that's on a form that is Maximized. Anyone know why I'm getting this problem? Any help would be appreciated.

I've since used Embedded C++ to access this API directly and do not get an error but the performance is terrible. I'm not a C++ developer so would like to get the managed code working. Am I not setting correct properties in the Managed code? I noticed the native code fails if the file already exists by returning an E_INVALID_ARGS.

I'm now looking at DirectShow to do this. Is it well supported? I would like to know why the Managed code isn't working but I need a solution quick. Is there another way of getting at the general camera API apart from what I mentioned here?

Thanks