

Re: Bitmap constructor throwing System.Exception

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.compactframework/2004-06/1357.htm>

From: Rick Winscot (*rickly_at_zyche*)

Date: 06/21/04

Date: Mon, 21 Jun 2004 02:51:51 -0600

Alex,

I think that this information is useful to the group... therefore:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/dotnetgcbasics.asp>

If your hope is to prevent the absorption of resources... your suggestion of calling Dispose() repeatedly may actually cause more problems if the users process is object intensive. From the MSDN document... "First, objects that need finalization live longer than objects that do not. In fact, they can live a lot longer." Given this... the virtual nature of memory on the PPC and potential of running out of -heap- (cough cough) space there is a possibility that long lived objects may be eating up available resources. Without some code to look at... it's all assumptions.

In regard to your correction on the disposing of objects. A manual call to Dispose() obviates the need for the collector to keep the object alive and call the finalizer... which may or may not be beneficial to this users implementation. I am curious... perhaps you can shed some light on how Dispose() is **not** involved in the collection process?

For your second correction... I made no representation that this problem was associated with managed or unmanaged resources. What I did suggest is that it would be helpful to see exactly what the user is doing in order to provide a factual basis for determination of a solution... which **is** the essence of Occam Razor's Principle.

While your comments revolve around the most common problem... and delve into great detail – I must remind you that what we are here to do, is to identify what the problem is... and how to effectively **fix** it.

#1 – it will be useful to many others to see the code... and solution (what we are all here for)

#2 – the information you provide falls easily into the trivia category instead of the solution toolkit as it provides no code reference to

individuals who need it...

#3 – if you feel that the solution to this common problem needs greater publicity... post a solution

Rick Winscot -> m_iPosts += 1;
www.zyche.com

"Alex Feinman [MVP]" <public_news@alexfeinman.com> wrote in message news:exBPFxVEHA.3596@tk2msftngp13.phx.gbl...

> *Seeing the code is always useful in troubleshooting. This particular problem*

> *however is rather commonly caused by not deallocating *unmanaged* resources.*

> *If you would care to review the Image.Dispose() method internals (appended),*

> *you would see that one thing it does is invoking AG.GL::Destroy on the m_how*

> *which is a pseudo-handle to the unmanaged part of the image. Internally this*

> *handle holds a real handle to a DIB (24 bit – ergo, large) and a DC where*

> *DIB is selected. While you are undoubtedly correct in your explanation of*

> *how GC collects the objects and invokes finalizers, you are wrong on*

> *2 counts. First, disposing the object does not mark it for collection.*

> *Deleting all references to it does. Second, I never said that calling*

> *Dispose will cause he managed memory to be released, nor I implied that the*

> *problem is caused by the lack of managed memory. Most often when loading*

> *large bitmaps, the unmanaged memory into which DIBs are loaded gets*

> *exhausted and causes exceptions when the underlying CreateDIBSection fails*

> *for the lack of memory*

>

> *Since the scenario I described is by far the most common cause for the error*

> *reported by the OP, I thought that acting on Occam razor principle it might*

> *make sense to try and eliminate it first.*

>

> *As for the comment on the "filling daily post quota" thing, I'll pretend I*

> *didn't see that.*

>

> --

> Alex Feinman

> ---

> Visit <http://www.opennetcf.org>

>

> "Rick Winscot" <rickly@zyche dot com> wrote in message

> news:ey7UP\$vVEHA.264@TK2MSFTNGP10.phx.gbl...

> > Alex,

> >

> > *Rather than make guesses... it would be far better to see some code and*

> > *verify what exactly is going on. If you will notice in his original*

> > *description – he mentioned that the emulator worked just fine while a*
> > *'custom' Win CE 5.0 configuration crashes.*
> >
> > *Also – please take a moment to review your information about .net*
garbage
> > *collection. Disposing any object merely *marks* it for collection... and*
> > *does not force the garbage collector to spin through unused objects at*
> > *'your' will. It is recommended to use GC.Collect() to cause the GC to*
run
> > *each unused objects finalizer. However, this process is not immediate*
> > *(instantaneous) either... and should be used only when necessary. There*
is
> > *a*
> > *high amount of probability that this users problem may stem from an err*
in
> > *logical approach... which can be easily adjusted for better performance.*
> > *It*
> > *is also possible that this problem may be a valid bug with Win CE 5.0.*
At
> > *this stage there isn't enough data to make any assumptions.*
> >
> > *I know that it is important to fill your daily post quota... but lets*
take
> > *discussions in a direction that lead to the best possible answer for*
each
> > *individual need. When source is needed... lets wait to see it before*
> > *anyone*
> > *invests any time in troubleshooting – in the end, it only wastes your*
time
> > *and theirs.*
> >
> > *Rick Winscot*
> > *www.zyche.com*
> >
> >
> > *"Alex Feinman [MVP]" <public_news@alexfeinman.com> wrote in message*
> > *news:%23ezVx\$tVEHA.3988@tk2msftngp13.phx.gbl...*
> > > *Are you disposing of the bitmap? By the time GC gets around to freeing*
the
> > > *object it might be too late for your app, so once you are done with*
the
> > > *bitmap, don't forget to invoke .Dispose on it*
> > >
> > > *--*
> > > *Alex Feinman*
> > > *---*
> > > *Visit <http://www.opennetcf.org>*
> > > *<ongg@w-linknospam.net> wrote in message*
> > > *news:eJZgfHoVEHA.3988@tk2msftngp13.phx.gbl...*
> > > > *I have an app that does a new Bitmap(filename); The filename exists*
for

>>>> *sure.*
>>>>
>>>> *I ran this from Visual Studio .Net 2003 in the emulator, and it works*
>>> *fine.*
>>>> *However, when I run this on a custom Windows CE 5.0 configuration ,*
I
>>>> *sometimes get a System.Exception.*
>>>>
>>>> *It has nothing to do with the file.. It seems like maybe after a*
> *certain*
>>>> *number of times of loading a bitmap, I get this error. (although*
that
>> *may*
>>>> *not be it).*
>>>>
>>>> *Has anyone seen this? It's a generic System.Exception, not a*
derived
>>> *class*
>>>> *(at least according to the Message property).*
>>>>
>>>> *Thanks!*
>>>> *Jay*
>>>>
>>>>
>>>
>>>
>>
>>
>
>