

Re: An interesting/annoying problem involving callbacks and native C and C#

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-02/msg01658.html>

- *From:* "Willy Denoyette [MVP]" <willy.denoyette@xxxxxxxxxxx>
 - *Date:* Tue, 13 Feb 2007 12:12:09 +0100
-

"per9000" <per9000@xxxxxxxxxxx> wrote in message
news:1171359396.043432.268740@xx

Hi Willy,

thanks for the hint on using stdcall, but it seems I have added that it to the real-life problem.

My real life problem is pretty much a mix of .NET C# and MC++; and native C and Fortran. I have methods written in C# that are stored in structs (delegates to them anyway) in C (from objects in C#). C and Fortran talks a lot and runs the methods from time to time given lots of conditions.

I've had two major problems. The first one is the one you mention, that is fixed by GC.KeepAlive(..), the typical exception was Unhandled Exception: System.AccessViolationException.

The other one, that I suspected was caused by a moved delegate, is that my program just shuts down – no exception or anything. A major problem for me here is if this is caused by some old exit condition in some obscure Fortran code that no one has touched for years or is it caused by the delegate that has moved. My idea was the following: IF the delegate has moved THEN any call to it (to some random memory I guess) could do about anything – for example shut down. Also this behavior only occurred after several seconds (almost a minute) or loops that really torture my native code so I suspected the garbage man.

In a blog at <http://blogs.msdn.com/sebby1234/archive/2006/04/05/565090.aspx> I found (under Pinned object): "The garbage collector has the possibility of physically moving the objects for which it is responsible..." This is pretty much why I started suspecting the garbage man.

So, my problem distills to a number of questions:
– When, why and how is the garbage man moving objects?

Re: An interesting/annoying problem involving callbacks and native C and C#

- Is pinning the answer to my prayers? Why/Why not?
- Can I "pin" managed objects? How/Why not?

Of course, any other help is also useful,

thanks,

Per

Moving delegates can never be a problem, so there is no need to pin, all you need to care about is the life-time of your delegate(s), otherwise said, you need to make sure the delegate(s) doesn't get GC'd. Make also sure that the "containers" in which you store the delegate references don't get collected, this would also cause the delegates to get collected (assumed there are no other references to the delegates). Note also that what's passed to unmanaged code is a "function pointer", the function it points to never moves (and doesn't go away with the delegate), code belongs to the 'type', it's not part of the object!.

Willy.