

Re: Inheritance problem with BinaryWriter

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2004-05/1821.html>

From: Jay B. Harlow [MVP – Outlook] (*Jay_Harlow_MVP_at_msn.com*)

Date: 05/08/04

Date: Sat, 8 May 2004 09:30:21 -0500

Eugene,

First question: What are you really attempting to do??

The Binary Writer (underlying stream really) is able to buffer the write requests for you, why are you putting the values in a Queue? Especially a queue that holds the position to write to! If you call write with 5 bytes & then call Flush, I would expect all 5 bytes are going to be at position 0 as you have not "actually" written anything yet, which totally doesn't make sense!!

Second question, can you post your complete source & sample code that fully demonstrates the problem, I am not able to recreate the problem.

Third: You do realize That calling MyBase.BaseStream forces the writer to be flushed, so as to ensure the Position & other properties of the Stream returned is valid (avoiding my comment from my first question). This indirect flushing of the basestream causes your writer to be flushed, causing QueuedBinaryWriter.Write to be called... Which causes your "infinite loop"

> *Any suggestions on how to solve this problem?*

Does your QueuedBinaryWriter have to inherit from BinaryWriter? Can it contain a BinaryWriter instead?

> *Public Class QueuedBinaryWriter*

```
Private Readonly m_writer As BinaryWriter
Private Readonly m_queue As Queue
```

```
Public Sub Write(value As Byte)
    m_Queue.Enqueue(New WriteRequest(MyBase.BaseStream.Position,
Value))
End Sub
```

```
Public Sub Write(value As Char)
    m_Queue.Enqueue(New WriteRequest(MyBase.BaseStream.Position,
Value))
```

End Sub

Alternatively write Enqueue WriteRequest so it does not rely on MyBase.BaseStream.

Hope this helps

Jay

"Eugene" <jsmith@aol.com> wrote in message
news:c7h4h8\$qtj\$1@news.epidc.co.kr...

```
> I'm trying to write a class which uses BinaryWriter as its base but allows  
> for queuing of write requests  
> Public Class QueuedBinaryWriter  
> Inherits BinaryWriter  
>  
> I override all the Write methods like so:  
> Public Overloads Overrides Sub Write(ByVal Value As Byte)  
> m_Queue.Enqueue(New WriteRequest(MyBase.BaseStream.Position,  
Value))  
> End Sub  
> 'same for all other Write variants  
>  
> I also add an overloaded Write method for my own datatype  
> private Overloads Sub Write(ByVal Request As WriteRequest)  
> Dim t As Type = Request.Type  
>  
> If t Is GetType(Boolean) Then  
> MyBase.Write(DirectCast(Request.Value, Boolean))  
> ElseIf t Is GetType(Byte) Then  
> MyBase.Write(DirectCast(Request.Value, Byte))  
> ' etc...  
> end sub  
>  
> and finally I override the Flush() method  
> Public Overrides Sub Flush()  
> For Each Request As WriteRequest In m_Queue  
> Seek(CInt(Request.Offset), SeekOrigin.Begin)  
> Write(Request)  
> Next  
> MyBase.Flush()  
> m_Queue.Clear()  
> End Sub  
>  
> The problem is in the MyBase.Write() call in the Write(WriteRequest)  
method.  
> It doesn't call the base class Write but instead calls  
> QueuedBinaryWriter.Write(). This leads to an infinite loop since each  
> QueuedBinaryWriter.Write() enqueues another WriteRequest.  
>  
> I checked the IL:  
> IL_0009: ldtoken [mscorlib]System.Boolean
```

> *IL_000e: call class [mscorlib]System.Type*
> *[mscorlib]System.Type::GetTypeFromHandle(valuetype*
> *[mscorlib]System.RuntimeTypeHandle)*
> *IL_0013: bne.un.s IL_0031*
> *IL_0015: ldarg.0*
> *IL_0016: ldarg.1*
> *IL_0017: ldfld object*
> *QueuedBinaryWriter.ciloci.QueuedBinaryWriter/WriteRequest::Value*
> *IL_001c: unbox [mscorlib]System.Boolean*
> *IL_0021: ldoobj [mscorlib]System.Boolean*
> *IL_0026: call instance void*
> *[mscorlib]System.IO.BinaryWriter::Write(bool)*
>
> *Any suggestions on how to solve this problem?*
>
> *Thank you for your time.*
>
>
>
>
>