

Re: Me vs. MyClass

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2005-09/msg00596.html>

- *From:* david <david@xxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 05 Sep 2005 22:26:23 -0700
-

On 2005-09-06, Jay B. Harlow [MVP - Outlook] <Jay_Harlow_MVP@xxxxxxxxxxxxxxxx> wrote:

> David,
>| Something else occurred to me: calling overridable functions from a
>| ctor. Seems pretty obscure though, and again it's providing language
>| support for a bad practice.
>| puzzled look ?
>
> C# & I believe both Java & C++ all support calling overridable methods in
> constructors.

I should have been clearer. These languages all do this, but they do it differently. As you point out, in .Net the vtable is built before the derived classes have run their ctor, so you have the possibility of an derived instance method called before the derived instance ctor.

In C++, the behavior is the opposite. The vtable for the derived class is built during construction of the derived class, so a call to a virtual method during a base class ctor calls the base class. The contract of "no instance method is called before construction" holds true.

MyClass could be used to emulate the C++ behavior. Is that why it's there? Probably not, I'm really grasping at straws. My best guess at this point is that its existence is related to the fact that vbc seems to treat all instance calls as virtual. But I'm probably overthinking it, it could just be that it was a cheap keyword to implement and somebody thought it might be useful someplace.

BTW, Java acts like C#/VB in this case. I think it's kind of a GC thing.

> In fact System.Windows.Forms.Form.CreateParams is called from the
> constructor! Which is why CreateParams cannot rely on fields initialized in
> the constructor as the base constructor calls it before the derived
> constructor has a chance for it to run.
>
> I really don't see how MyClass adds or takes away support to calling
> overridable methods in constructors...
>

Re: Me vs. MyClass

> Of course calling an overridable method in a constructor needs to be done
> with caution.
>
> Hope this helps
> Jay
>
>
> "david" <david@xxxxxxxxxxxxxxxxxxxx> wrote in message
> <news:slrndhpm51.hs5.david@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
> | On 2005-09-05, Jay B. Harlow [MVP – Outlook]
> <Jay_Harlow_MVP@xxxxxxxxxxxxxxxxxxxx> wrote:
> | > David,
> | > As I stated:
> | >
> | > It appears we may need to agree to disagree on MyClass & constructors...
> | > :-|
> | >
> | > Agreed.
> | >
> | > If I find additional info on a viable reason to call MyClass outside of
> | > constructors I will post it.
> | >
> | > Something else occurred to me: calling overridable functions from a
> | > ctor. Seems pretty obscure though, and again it's providing language
> | > support for a bad practice.
> | >
> | ><<snip>>
> | >
> | >
> | >

• References:

- ◆ [Me vs. MyClass](#)
 ◇ From: Water Cooler v2
- ◆ [Re: Me vs. MyClass](#)
 ◇ From: Sathyaish
- ◆ [Re: Me vs. MyClass](#)
 ◇ From: dfoster
- ◆ [Re: Me vs. MyClass](#)
 ◇ From: Jay B. Harlow [MVP – Outlook]
- ◆ [Re: Me vs. MyClass](#)
 ◇ From: dfoster
- ◆ [Re: Me vs. MyClass](#)
 ◇ From: Jay B. Harlow [MVP – Outlook]
- ◆ [Re: Me vs. MyClass](#)
 ◇ From: dfoster
- ◆ [Re: Me vs. MyClass](#)
 ◇ From: Jay B. Harlow [MVP – Outlook]
- ◆ [Re: Me vs. MyClass](#)

Re: Me vs. MyClass

◇ *From:* david

◆ ***Re: Me vs. MyClass***

◇ *From:* Jay B. Harlow [MVP – Outlook]

◆ ***Re: Me vs. MyClass***

◇ *From:* david

◆ ***Re: Me vs. MyClass***

◇ *From:* Jay B. Harlow [MVP – Outlook]

- Prev by Date: ***Re: Sort an Array List***
- Next by Date: ***Querying a dataset***
- Previous by thread: ***Re: Me vs. MyClass***
- Next by thread: ***Re: Me vs. MyClass***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***