

ECMA Wrong – Class and Object Initialization Rules – Help!

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.clr/2005-04/msg00078.html>

- *From:* George <George@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 12 Apr 2005 03:25:03 -0700
-

Hi,

ECMA states in Section 1.8.1.4 (Partition III) on Class and Object Initialization Rules:

"An object constructor shall not return unless a constructor for the base class or a different construct for the object's class has been called on the newly constructed object. The verification algorithm shall treat the this pointer as uninitialized unless the base class constructor has been called."

This means in particular that, if a .ctor didn't invoke a base class .ctor but another .ctor of the same class, then the "this" pointer is not initialized. For example, class A has 2 .ctors that invoke eachother: recursive constructor invocation is allowed at the bytecode level!

BUT, the examples I tried show that the "this" pointer is initialized even after invoking another .ctor of the same class – for example I can load the "this" into a local variable.

SO, it seems to me that the "this" is considered initialized also after invoking a ctor of the same class and NOT ONLY a ctor of the base class!

=====

There is another issue which is not clear.

Looking in the ROTOR source, I found the following commented idea:

"If there are any locals (or arg slot 0) containing uninit vars, it is illegal to be in a try block."

This corresponds to error "VER_E_THIS_UNINIT_EXCEP" "Uninitialized this on entering a try block"

Beside the fact that this is not documented in ECMA, I cannot have it confirmed by examples.

Is this verification error still checked in the current version of the verifier?

Many thanks for any help!

George

• ***Follow-Ups:***

- ◆ ***Re: ECMA Wrong – Class and Object Initialization Rules – Help!***
 ◇ *From:* Matt Grice [MSFT]
- ◆ ***Re: ECMA Wrong – Class and Object Initialization Rules – Help!***
 ◇ *From:* Daniel O'Connell [C# MVP]
- ◆ ***RE: ECMA Wrong – Class and Object Initialization Rules – Help!***
 ◇ *From:* Ravichandran J.V.
- ◆ ***RE: ECMA Wrong – Class and Object Initialization Rules – Help!***
 ◇ *From:* Ravichandran J.V.

- Prev by Date: ***Re: How do I code a Module Constructor in C# or C++?***
- Next by Date: ***RE: AutoUpdate WinformApp (AppDomain.ShadowCopyFiles default setting)***
- Previous by thread: ***How do I code a Module Constructor in C# or C++?***
- Next by thread: ***RE: ECMA Wrong – Class and Object Initialization Rules – Help!***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***