INFO: Workaround for a License Compiler Exception When Targeting x64 in .NET Framework 4.0 and up

We have found that there is a bug in Visual Studio when it comes to building a 64-bit application on a 64-bit system. Visual Studio will use the 32-bit version of license compiler (LC.exe) when the build configuration is set to x64.

Special Note

Visual Studio 2022 is now a 64 bit process .. If you are getting errors about incompatible format in VS2022, the solution would be to target x64 and user our x64 DLLs

Support recommends not applying this fix unless you are explicitly running into issues where embedding license in 64 bit apps is failing but works when targeting x86) in VS2019 or older

For issues in VS2022, please consider creating a support case, or calling in to support.

Main Article ... continued

***Please note, this article refers to the .NET Framework 4.0 and up ... for correct information regarding the .NET Framework 2.0, 3.0 and 3.5, please refer to* INFO: Workaround for a license compiler exception when targeting x64 in .NET Framework 2.0, 3.0, or 3.5**

To workaround this issue you will need to modify your application project file in a text editor. Add the highlighted line shown below to all of your 64-bit build configurations (Note: the code below is an EXAMPLE - you need to specify the actual valid directory for a 64 bit version of lc.exe):

```
<CodeAnalysisUseTypeNameInSuppression>true</CodeAnalysisUseTypeNameInSuppression>

<CodeAnalysisModuleSuppressionsFile>GlobalSuppressions.cs</CodeAnalysisModuleSuppressionsFile>
  <ErrorReport>prompt</ErrorReport>
  <LCToolPath>C:\Program Files (x86)\Microsoft SDKs\Windows\v8.0A\bin\NETFX 4.0 Tools\x64</LCToolPath>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|x64' ">
  <OutputPath>bin\x64\Release\</OutputPath>
  <DefineConstants>TRACE</DefineConstants>
  <Optimize>true</Optimize>
  <DebugType>pdbonly</DebugType>
  <PlatformTarget>x64</PlatformTarget>

<CodeAnalysisUseTypeNameInSuppression>true</CodeAnalysisUseTypeNameInSuppression>

<CodeAnalysisModuleSuppressionsFile>GlobalSuppressions.cs</CodeAnalysisModuleSuppressionsFile>
  <ErrorReport>prompt</ErrorReport>
  <LCToolPath>C:\Program Files (x86)\Microsoft SDKs\Windows\v8.0A\bin\NETFX 4.0 Tools\x64</LCToolPath>
</PropertyGroup>
```

This will tell Visual Studio where to get the license compiler during a build.

Directories

NOTE that the example above uses the LCToolPath of C:\Program Files (x86)\Microsoft SDKs\Windows\v8.0A\bin\NETFX 4.0 Tools\x64

INFO: Workaround for a License Compiler Exception When Targeting x64 in .NET Framework 4.0 and up

Depending on target framework and version the path may differ .. but the gist here is that you need to find the path to a 64 bit version of lc.exe

To find what you have available, you can do the following:

Open a windows Command Prompt or PowerShell

cd C:\Program Files (x86)\Microsoft SDKs\Windows\

dir /s /b lc.exe

note that it may well return multiple versions - not all are usable here.. what you need is to find one in an x64 subdirectory which is at or below the target framework you need (so if you're targeting .NET 4.6.1 you could use the one for .NET 4.0, .NET 4.5.1, .NET 4.5.2, .NET 4.6, or .NET 4.6.1 but may run into trouble if you target .NET 4.6.2 or higher...

here is the output from my command

Microsoft Windows [Version 10.0.19043.1348](c) Microsoft Corporation. All rights reserved.

C:\windows\System32>cd C:\Program Files (x86)\Microsoft SDKs\Windows\

C:\Program Files (x86)\Microsoft SDKs\Windows>dir /s /b lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6 Tools\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6 Tools\x64\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.1 Tools\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.1 Tools\x64\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.2 Tools\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.2 Tools\x64\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8 Tools\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8 Tools\x64\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin\x64\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v8.0A\bin\NETFX 4.0 Tools\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v8.0A\bin\NETFX 4.0 Tools\x64\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v8.1A\bin\NETFX 4.5.1 Tools\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v8.1A\bin\NETFX 4.5.1 Tools\x64\lc.exe

C:\Program Files (x86)\Microsoft SDKs\Windows>

So, as you can see, it found several.. but not all of them are in \x64\ subdirectories. The only paths we would want would be those with x64

Also, remember, the path above does not want the actual lc.exe it wants the containing path.. so we're left with these options:

C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin\x64

C:\Program Files (x86)\Microsoft SDKs\Windows\v8.0A\bin\NETFX 4.0 Tools\x64

C:\Program Files (x86)\Microsoft SDKs\Windows\v8.1A\bin\NETFX 4.5.1 Tools\x64

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6 Tools\x64

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.1 Tools\x64

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.2 Tools\x64

C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8 Tools\x64

Now, the 7.0A is a .NET 4.0 tool (I happen to know this) but if you were unsure, just go with one from the NETFX x.y Tools paths (NETFX is Microsoft-speak for .NET Framework and the paths that are thus labeled are easier to find...)

So in my example above, I used

<LcToolPath>C:\Program Files (x86)\Microsoft SDKs\Windows\v8.0A\bin\NETFX 4.0 Tools\x64</LcToolPath>

as this should in theory work for any .NET 4.x situation... if I ran into trouble I would try upping to

C:\Program Files (x86)\Microsoft SDKs\Windows\v8.1A\bin\NETFX 4.5.1 Tools\x64

INFO: Workaround for a License Compiler Exception When Targeting x64 in .NET Framework 4.0 and up

For more information, please see:

INFO: Bitness Roundup Whitepaper: x86, x64, AnyCPU

Original Article

Q10341 - INFO: Workaround for a license compiler exception when targeting x64 in .NET Framework 4.0 and up

Last Update

2026-10-06 - TD - (Minor formatting updates)

Atalasoft Knowledge Base
https://www.atalasoft.com/kb2/KB/50145/INFO-Workaround-for-a-License-Compil...