# ChilKat 64-Bit Issue

**April 5, 2009**

[**Chilkat .NET for x64 (64-bit Windows)**](http://www.cknotes.com/?p=165)

Filed under: [x64](http://www.cknotes.com/?cat=93) — Tags: [64-bit](http://www.cknotes.com/?tag=64-bit), [error messages](http://www.cknotes.com/?tag=error-messages) — admin @ 10:30 am

The **“An attempt was made to load a program with an incorrect format.”** indicates that your program attempted to load a win32 assembly on an x64 Windows OS, or the reverse.

There is a separate Chilkat .NET assembly also named ChilkatDotNet2.dll for 64-bit Windows.  It may be downloaded at: <http://www.chilkatsoft.com/downloads.asp> This is for the 2.0/3.5 (or greater) Framework.  There is no x64 support for the .NET 1.0/1.1 Framework.

If you are developing on a 32-bit platform, but you need to deploy to a 64-bit platform, make sure that Visual Studio does not automatically copy the win32 ChilkatDotNet2.dll to the x64 target computer.  Your project may select “Any CPU” for it’s target platform, but you must deploy the correct ChilkatDotNet2.dll to the target computer.

**Important:** The 32-bit ChilkatDotNet2.dll can actually be used on both win32 and x64 systems, however, the EXE that loads the DLL (i.e. assembly) must be running as a 32-bit app. If you write a .NET application and target “Any CPU”, then it will run as a 64-bit application on an x64 computer, and therefore the x64 build of ChilkatDotNet2.dll would be required. However, if you target “x86″ instead of “Any CPU”, then your application will run as a 32-bit app and the win32 build of ChilkatDotNet2.dll can be used on both 32-bit and 64-bit systems.

**Beware of Version Mismatch**: The version of the x64 Chilkat assembly must also match the version referenced by your Visual Studio project.  Otherwise you’ll get an assembly load-time error indicating a version mismatch.

**Conditional Deployment:** The Visual Studio Setup & Deployment project (.msi) is capable of conditional deployment.  This means you may include both win32 and x64 versions of ChilkatDotNet2.dll in your .msi, and the correct one is installed depending on the platform where it is run.  For more information:  [MSI Conditional Deployment](http://www.chilkatsoft.com/p/p_541.asp).

**Windows Services**: When using ChilkatDotNet2.dll from a Windows Serivce, the Chilkat assembly may need to be installed in the Global Assembly Cache (GAC).  This this blog post:  [Using Chilkat x64 .NET Assembly in a Windows Service](http://www.cknotes.com/?p=95).

**ClickOnce Deployment:** See this blog post about [x64 ClickOnce deployment](http://www.cknotes.com/?p=212).

[TrackBack URL](http://www.cknotes.com/wp-trackback.php?p=165)

April 17, 2008

# Conditional Deployment: x64 vs win32

A Visual Studio Setup and Deployment project can be used to create a .msi installer the conditionally deploys different files based on the Intel processor architecture at runtime (when the installer runs).

The **Condition** property may be set for each file added to the Setup and Deployment project. To set the condition so that a file only installs on x64 computers, set the Condition property to "**Intel=9**" (without the quotes). To only install a file on win32 systems, set the Condition property to "**Intel=0**".

[**ChilkatDotNet2.dll on x64 from a Windows Service**](http://www.cknotes.com/?p=95)

Filed under: [error messages](http://www.cknotes.com/?cat=23" \o "View all posts in error messages), [x64](http://www.cknotes.com/?cat=93) — Tags: [64-bit](http://www.cknotes.com/?tag=64-bit), [GAC](http://www.cknotes.com/?tag=gac), [Service](http://www.cknotes.com/?tag=service), [x64](http://www.cknotes.com/?tag=x64) — admin @ 11:30 am

Here’s the secret to getting the .NET runtime to load the ChilkatDotNet2.dll from a Windows Service: It must be installed in the GAC, but there’s a separate GAC for 64-bit assemblies:

(with permission from the Chilkat customer)

I was able to find a solution to this. I'm running a 64-bit version of

Vista, and there turns out to be a separate GAC for 64-bit assemblies, and

my application was searching there, and was not able to find the 32-bit

Chilkat assembly. I installed the 64-bit version in addition to the 32-bit

one in the GAC, and then the assembly loaded fine. Why this works when

running as a normal application but not when running as a service I don't

know, but at least it is working now.

Thanks for your excellent support!

This was the original error:

(with permission from the Chilkat customer)

> > Thanks for your immediate response. I tried to install the assembly in the

> > GAC, but it does not seem to help. I am able to load another GAC assembly

> > without problems:

> >

> > Assembly a = Assembly.Load("Microsoft.Practices.EnterpriseLibrary.Logging,

> > Version=2.0.0.0, Culture=neutral, PublicKeyToken=bf81d2044fa6b642");

> > Assembly b = Assembly.Load("ChilkatDotNet2, Version=8.7.0.0,

> > Culture=neutral, PublicKeyToken=eb5fc1fc52ef09bd");

> >

> > The first load does not throw an exception, but the second does:

> >

> > System.IO.FileNotFoundException: Could not load file or assembly

> > 'ChilkatDotNet2, Version=8.7.0.0, Culture=neutral,

> > PublicKeyToken=eb5fc1fc52ef09bd' or one of its dependencies. The system

> > cannot find the file specified.

> >

> > When turning on logging of the binding, it does not seem like the GAC is

> > searched at all, only the file system (current directory).

> >

> > I'm suspecting that this is a permission problem, but running the service as

> > LocalSystem or LocalService gives the same result.