**[Marshalling .NET Object when class is in another namespace](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace)**

Top of Form

**No problem. We won't show you that ad again. Why didn't you like it?**

* Uninteresting
* Misleading
* Offensive
* Repetitive

[Oops! I didn't mean to do this.](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace)

Bottom of Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote1down vote[favorite](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace)  **1** | I am having a problem with Marshalling an object across application domains in a .NET Windows Service.  I have created two app's that marshal an object across an app domain and run code through a proxy MarshalByRefObject  The first application was a simple proof of concept since I do not have a lot of experience with marshalling across application domains. It contains a single project where the MarshalByRefObject is defined in the same namespace as the rest of the project. This app works fine, using almost the same code as my other app. The main difference is that the class I am Marshalling in the second app is defined in another Namespace.  The other project is more complex, it is a Windows Service with multiple projects. The main Windows Service loads a library that that does the marshalling. The class type of the Marshal target type is defined in another library, so I use the fully qualified namespace/class name.  The problem I am facing is that when it gets to the last line of the code below it throws an exception:  **Could not load CompanyName.ProductGroup.BusinessObjects.ProductName.MarshalByRefScriptCompiler from assembly ProductNameService** where product name is the main Windows Service class.  The Code:  AppDomain compilerDomain = null;  AppDomainSetup compilerDomainSetup;  CompanyName.ProductGroup.BusinessObjects.ProductName.MarshalByRefScriptCompiler scriptCompiler;  ...  // Setup a seperate AppDomain  compilerDomainSetup = new AppDomainSetup();  exeAssembly = Assembly.GetEntryAssembly().FullName;  compilerDomainSetup.ApplicationBase = System.Environment.CurrentDirectory;  compilerDomainSetup.DisallowBindingRedirects = false;  compilerDomainSetup.DisallowCodeDownload = true;  compilerDomainSetup.ConfigurationFile = AppDomain.CurrentDomain.SetupInformation.ConfigurationFile;  compilerDomain = AppDomain.CreateDomain("LiveLinkCSScriptDomain", null, compilerDomainSetup);  // Create an instance of the MarshalByRefScriptCompiler in the other AppDomain  scriptCompiler = (CompanyName.ProductGroup.BusinessObjects.ProductName.MarshalByRefScriptCompiler)compilerDomain.CreateInstanceAndUnwrap(exeAssembly, typeof(CompanyName.ProductGroup.BusinessObjects.ProductName.MarshalByRefScriptCompiler).FullName);  I have done research on this exception and almost everything I find says that it is a problem with versioning of the DLL, however my DLL's are not in the GAC, and there are no other versions installed. I am doing a clean build and installing the service with installutil.  I used the [MSDN documentation](http://msdn.microsoft.com/en-us/library/system.appdomain.aspx) as a guide for creating the code that does the marshalling  I am wondering if there is an issue with loading the MarshalByRefScriptCompiler because the type is in another library. I am able to create a MarshalByRefScriptCompiler in a simple winforms app, but I get the exception in my windows service.  Any tips or insight would be greatly appreciated!  [c#](http://stackoverflow.com/questions/tagged/c%23) [.net](http://stackoverflow.com/questions/tagged/.net) [windows-services](http://stackoverflow.com/questions/tagged/windows-services) [marshalling](http://stackoverflow.com/questions/tagged/marshalling) [appdomain](http://stackoverflow.com/questions/tagged/appdomain)   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/q/7890230)|[improve this question](http://stackoverflow.com/posts/7890230/edit) | [edited Oct 25 '11 at 18:11](http://stackoverflow.com/posts/7890230/revisions)  [[https://www.gravatar.com/avatar/ca7873b97d8b19325cddbc1a73bc46e2?s=32&d=identicon&r=PG](http://stackoverflow.com/users/473921/russell-mcclure)](http://stackoverflow.com/users/473921/russell-mcclure)  [Russell McClure](http://stackoverflow.com/users/473921/russell-mcclure) **2,418**815 | asked Oct 25 '11 at 13:45  [[https://www.gravatar.com/avatar/17049ac98970bc77eefdf6418235b0cd?s=32&d=identicon&r=PG](http://stackoverflow.com/users/488004/dmck)](http://stackoverflow.com/users/488004/dmck)  [dmck](http://stackoverflow.com/users/488004/dmck) **3,408**11646 | |
|  | |  |  | | --- | --- | |  |  |   add comment |

**2 Answers**

[active](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace?answertab=active#tab-top)[oldest](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace?answertab=oldest#tab-top)[votes](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace?answertab=votes#tab-top)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote2down voteaccepted | I should be able to help you. I've spent many hours recently ([How do I pass references as method parameters across AppDomains?](http://stackoverflow.com/questions/2932370/how-do-i-pass-references-as-method-parameters-across-appdomains)) working on different cross appdomain marshaling problems. My first suggestion would be to try using **CreateInstanceFromAndUnwrap** instead of **CreateInstanceAndUnwrap**.  I'm also a little wary of this line:  compilerDomainSetup.ApplicationBase = System.Environment.CurrentDirectory;  How is your original AppDomain being created? Are you hosted in IIS, in which case your original AppDomain will be using ShadowCopy? Are all the dlls involved in a single folder?  EDIT:  To summarize, you can use **CreateInstanceAndUnwrap** if your **compilerDomainSetup.ApplicationBase** is set to the directory containing your dll and you pass in the correct first param (e.g. typeof(MarshalByRefScriptCompiler).Assembly.FullName).  Or, you can use **CreateInstanceFromAndUnwrap** and just pass in the location (e.g. typeof(MarshalByRefScriptCompiler).Assembly.Location) of the containing assembly as the first param.   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/7890480)|[improve this answer](http://stackoverflow.com/posts/7890480/edit) | [edited Oct 25 '11 at 16:59](http://stackoverflow.com/posts/7890480/revisions) | answered Oct 25 '11 at 14:02  [[https://www.gravatar.com/avatar/ca7873b97d8b19325cddbc1a73bc46e2?s=32&d=identicon&r=PG](http://stackoverflow.com/users/473921/russell-mcclure)](http://stackoverflow.com/users/473921/russell-mcclure)  [Russell McClure](http://stackoverflow.com/users/473921/russell-mcclure) **2,418**815 | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | Thanks for the response. I changed the ApplicationBase to: compilerDomainSetup.ApplicationBase = @"C:\ServiceSchedulerDebug\Debug"; which is where all the DLLs are (all in a single folder). However in the exception is still shows it looking in C:\Windows\System32. Any ideas?– [dmck](http://stackoverflow.com/users/488004/dmck)[Oct 25 '11 at 14:41](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace#comment9631366_7890480) | | |  |  | | --- | --- | |  |  | | Almost sounds like your change didn't get picked up. Your windows service is running in the C:\ServiceSchedulerDebug\Debug folder?– [Russell McClure](http://stackoverflow.com/users/473921/russell-mcclure)[Oct 25 '11 at 14:55](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace#comment9631637_7890480) | | |  |  | | --- | --- | |  |  | | Yes, I am doing a debug build and installing with installutil from the command line. The service is installed in C:\ServiceScheduler\Debug (changed the dir). AppDomain.CurrentDomain.BaseDirectory is C:\ServiceScheduler\Debug. I'm still trying to figure out why it's looking in C:\Windows\System32, I think this is causing the problem.– [dmck](http://stackoverflow.com/users/488004/dmck)[Oct 25 '11 at 15:48](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace#comment9632769_7890480) | | |  |  | | --- | --- | |  |  | | You need to use CreateInstanceFromAndUnwrap and pass in the path to the dll as the first param: typeof(MarshalByRefScriptCompiler).Assembly.Location– [Russell McClure](http://stackoverflow.com/users/473921/russell-mcclure)[Oct 25 '11 at 16:06](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace#comment9633142_7890480) | | |  |  | | --- | --- | |  |  | | Running a service through windows, the working directory is the windows\system directory. That's why you're seeing that behaviour.– [Chris Stavropoulos](http://stackoverflow.com/users/87733/chris-stavropoulos)[Oct 25 '11 at 18:10](http://stackoverflow.com/questions/7890230/marshalling-net-object-when-class-is-in-another-namespace#comment9635529_7890480) | |