

how to resolve Duplicate namespaces in c#

Asked 7 years, 10 months ago Active 7 years, 10 months ago Viewed 2k times



I have 2 assemblies Combres and log4net

1

Both assemblies contain the same log4net.Appender namespace (internal code included) - I need to inherit log4net.Appender.AdoNetAppender .



How do I accomplish this.



c#

.net

namespaces

edited Sep 6 '11 at 14:57

asked Sep 6 '11 at 14:49



David

3,222

12

34

58

using log4net.Appender.AdoNetAppender; doesn't get the right one for you? – Marco Sep 6 '11 at 14:50

@Marco: AdoNetAppender is a class. – Robert Harvey ♦ Sep 6 '11 at 14:52

1 You answered your question by inheriting log4net.Appender.AdoNetAppender. – saber Sep 6 '11 at 14:53

no it doesnt know which one to use cos the namespace is duplicated – David Sep 6 '11 at 14:53

I'm quite sure that Combres does **not** use the log4net.Appender namespace but Combres.Loggers namespace.... Maybe the root cause of the error is elsewhere – Stephan Bauer Sep 6 '11 at 15:04

4 Answers



Fully qualify the type. For example if you are trying to inherit from this class:

6

```
class MyAppender : log4net.Appender.AdoNetAppender
```

If you are trying to use / create an instance of this class:



```
var appender = log4net.Appender.AdoNetAppender;
```

If Combres and log4net both contain a type AdoNetAppender in the log4net.Appender namespace then you are in more trouble (and someone made a mistake - namespaces are designed to avoid these sorts of conflicts).

If this does happen then you can use the assembly reference "Aliases" property to resolve the conflict as described in [What use is the Aliases property of assembly references](#).

edited May 23 '17 at 12:19



Community ♦

1 1

answered Sep 6 '11 at 14:52



Justin

67.1k

42

192

331

awesome answer totally HG (Higher Grade) – David Sep 6 '11 at 15:10



You can probably specify an alias for one of the namespaces, like this:

7

```
using MyNameSpace = log4net.Appender;
```



Then inherit MyNameSpace.AdoNetAppender

answered Sep 6 '11 at 14:52



Michael Bray

12.1k

6

36

62



Take a look at the documentation for [extern alias](#). It allows you to explicitly reference classes in your code even if they are in the same namespace and have the same name.

2

You might, for example, reference the log4net classes like so:



```
extern alias l4n;
```

```
//... further down
```

```
l4n::log4net.Appender.AdoNetAppender l4nAppender = null;
```

The "ln" alias also has to be added to the property page for the DLL reference in Visual Studio.

answered Sep 6 '11 at 14:59



[Enigmativity](#)

81.1k 9 68 135

▲ I'm quite sure that Combres itself does not use the `log4net.Appender` namespace but `Combres.Loggers` namespace.

0

But `Combres` (2.2.1) contains a reference to `log4net`. If your application contains a reference to `log4net` as well (possibly even in another location than the `log4net` assembly that is referenced by `Combres`), this could result in the error you described.

▼ *(Multiple log4net-assemblies are referenced and of course these all contain the `log4net.Appender`)*

answered Sep 6 '11 at 15:12



[Stephan Bauer](#)

7,523 3 29 53