

Using Statements vs Namespace path? C#

Asked 8 years, 3 months ago Active 1 year, 7 months ago Viewed 6k times

▲ I recently stopped using `using-statement`s and instead use the full namespace path of any `.net` object that I call.

20 Example:

▼

```
using System;

namespace QuizViewer
{
    class Class1
    {
        Console.WriteLine("Hello World!");
    }
}
```

This is what I do now.

```
namespace QuizViewer
{
    class Class1
    {
        System.Console.WriteLine("Hello World!");
    }
}
```

Before you ask why I do this, I am using this style so that I can see exactly where my objects are coming from and it's easier when using the different Timer objects and other objects with similar names.

Is there any performance increase or decrease in this style of programming?

`c#` `.net` `namespaces` `using-statement`

edited Mar 19 '18 at 14:36



Nae

6,847 4 18 45

asked Jul 8 '11 at 18:07



Kyle Uithoven

1,324 4 22 41

- 11 I'd say that this style will lead to a significant decrease in the performance of the *person* who is reading or writing the code... (It's fair enough to do this for a few classes, such as `Timer`, where there are several equally-named classes, but for the most part, I'd consider the namespaces to be noise.) – [Aasmund Eldhuset](#) Jul 8 '11 at 18:10
- You can always hover over a type name in Visual Studio to see the full Namespace and Classname of the type. – [Kyle Trauberman](#) Jul 8 '11 at 18:12
- 4 Note that you are talking about *using directives*, not *using statements*. The *using statement* is the form `using(var stream = File.Open(...)) { ... }`. – [Eric Lippert](#) Jul 8 '11 at 18:38
- Note also that this related question might help you understand why there is no performance impact of this change: stackoverflow.com/questions/6614375/... – [Eric Lippert](#) Jul 8 '11 at 18:39
- 7 Finally, note that if you are eschewing "using" because of a specific confusion between two similarly-named things, you can use an *using alias directive*:
`using FrobTimer = BogoSoft.Froboznicator.Timer;` -- now you can use the identifier `FrobTimer` in that file and the compiler will know that you mean the fully-qualified type. – [Eric Lippert](#) Jul 8 '11 at 18:41

6 Answers



There is zero performance difference because the compiler ALWAYS puts in the full name - using is only a hint for the compiler, the runtime doesn't know or support that.

37



However, once you memorize where the objects come from you will look at this as silly and verbose. There is just so much noise and people just know that `Path` is from `System.IO`, `Console` is in `System` and `StringBuilder` is in `System.Text`.



One downside of your approach: Without using, no extension methods outside of the current namespace. Have fun writing `System.Linq.Enumerable.Where(inputSequence,...)` instead of just `inputSequence.Where(...) :`

answered Jul 8 '11 at 18:10



[Michael Stum](#) ♦

122k 103 370 515

For one `Where()`, it's still bearable. But a more complicated query would quickly become illegible. – [svick](#) Jul 9 '11 at 12:57



I think that this style result in a programmer performance decrease :). I use the using statement and usually it is clear from code to which namespace the class belong. If not, press F12.

6

Just my 2c.

answered Jul 8 '11 at 18:10



platon

4,975 1 16 22

In the case of this, how do you "tell" which timer it is before pressing F12? `System.Timers.Timer _timer; System.Threading.Timer _timer2; System.Windows.Forms.Timer _timer3;` – [Kyle Uithoven](#) Jul 8 '11 at 18:12 ✎

2 Hover over the `Timer` type name or an instance of it to see the full type name. – [Kyle Trauberman](#) Jul 8 '11 at 18:15

3 Just want to continue ... From my point of view, using `var` in C# everywhere instead of declaring a strong typed variable is really a bad style code... – [platon](#) Jul 8 '11 at 18:20

3 @platon using `var` in C# does not stop a variable being strongly typed. From [MSDN](#) - "It is important to understand that the `var` keyword does not mean "variant" and does not indicate that the variable is loosely typed, or late-bound. It just means that the compiler determines and assigns the most appropriate type." – [luketorjussen](#) Jul 8 '11 at 19:03

1 Indeed, I agree, I may be expressed myself badly. I do not like to see code where all the variables are declared as `var`. It is a nightmare for a programmer who did not write this code. It is awful to read the code written this way and I always have an idea to change it ... :-)) – [platon](#) Jul 8 '11 at 19:06 ✎

Short answer no: it is the same code that is compiled.

6

answered Jul 8 '11 at 18:10



Avada Kedavra

5,787 4 26 44

There's no performance impact; it's mostly a stylistic choice. I find that using `using` statements reduces the clutter. Plus, with Intellisense, it's easy enough to see the fully qualified namespace.

5

answered Jul 8 '11 at 18:10



Daniel Mann

43k 10 73 93



The only performance hit, is the hit you take to type it all out, and with you or others reading it.

5

Using statements are to help readability, not really for performance.



answered Jul 8 '11 at 18:10



[Rangoric](#)

2,354

1

14

16



If you disassemble both of this pieces of code and look at IL code, you'll find that compiler always references all the types by it's full names. That is absolutely identical ways to operating types.

0



answered Oct 28 '16 at 22:19



[Aave](#)

321

3

13