How do I use the C#6 “Using static” feature?

Asked 4 years, 2 months ago Active 6 months ago Viewed 45k times

113

22

I'm having a look at a couple of the new features in C# 6, specifically, *"using static"*.

> ***using static*** *is a new kind of using clause that lets you import static members of types directly into scope.*
>
> (Bottom of the blog post)

The idea is as follows, according to a couple of tutorials I found,
Instead of:

```
using System;

class Program
{
    static void Main()
    {
        Console.WriteLine("Hello world!");
        Console.WriteLine("Another message");
    }
}
```

You can omit the repeated `Console` statement, using the new C# 6 feature of using static classes:

```
using System.Console;
//           ^ `.Console` added.
class Program
{
    static void Main()
    {
        WriteLine("Hello world!");
        WriteLine("Another message");
    } // ^ `Console.` removed.
}
```

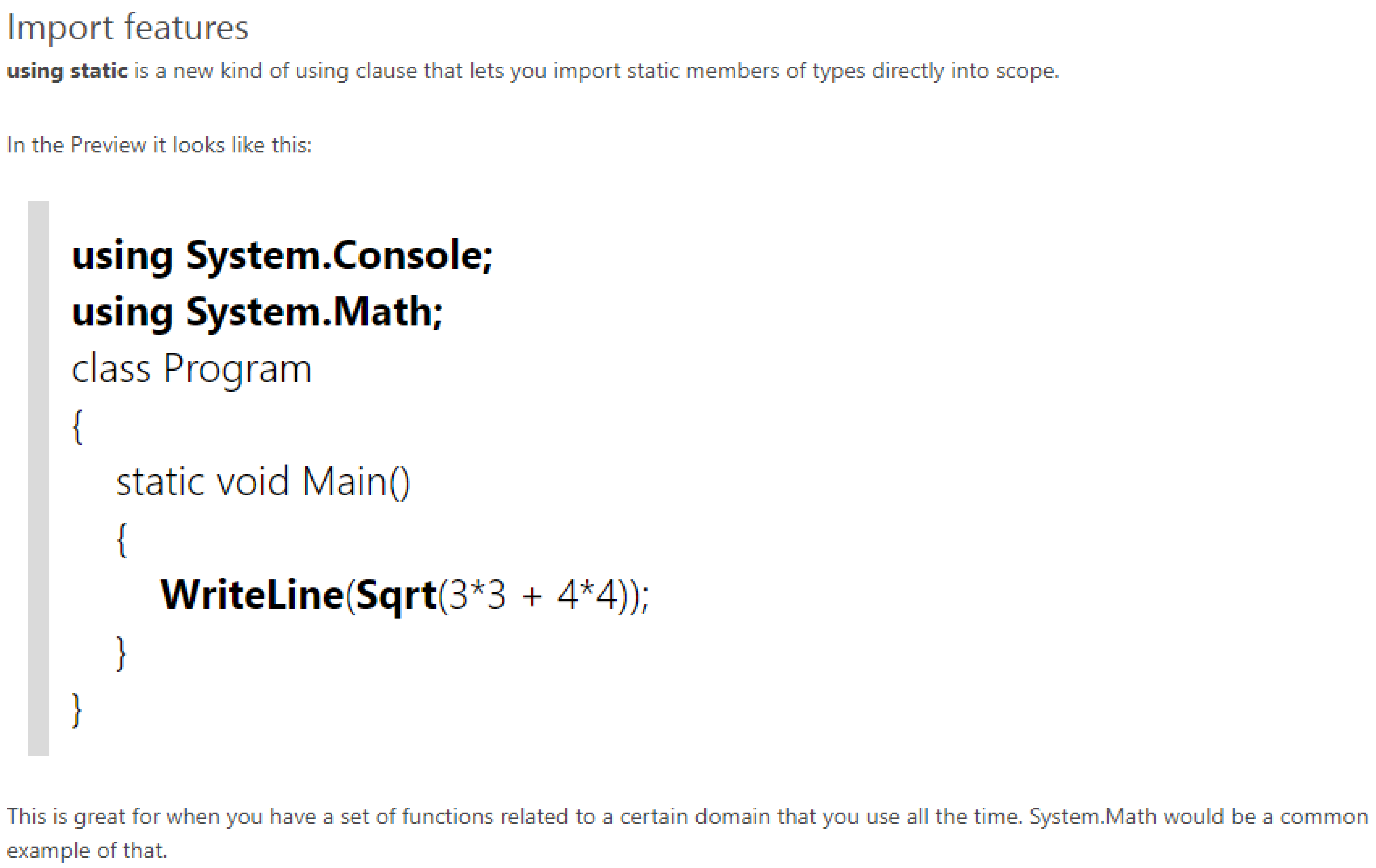
However, this doesn't appear to be working for me. I'm getting an error on the `using` statement, saying:

> *"A '`using namespace`' directive can only be applied to namespaces; '`Console`' is a type not a namespace. Consider a '`using static`' directive instead"*

I'm using visual studio 2015, and I have the build language version set to *"C# 6.0"*

What gives? Is the msdn blog's example incorrect? Why doesn't this work?

The blog post has now been updated to reflect the latest updates, but here's a screenshot in case the blog goes down:



Import features

using static is a new kind of using clause that lets you import static members of types directly into scope.

In the Preview it looks like this:

```
using System.Console;
using System.Math;
class Program
{
    static void Main()
    {
        WriteLine(Sqrt(3*3 + 4*4));
    }
}
```

This is great for when you have a set of functions related to a certain domain that you use all the time. System.Math would be a common example of that.

> *Note: The Preview will only import members of static classes. We have since changed the design in several ways:*
>
> 1. *The syntax will be more different from current using clauses – it will have the keywords "using static".*
> 2. *Members of non-static types can be imported, including structs and en ums*
> 3. *Members of VB modules and F# top level functions can be imported*
> 4. *Extension methods will not be imported into top-level scope; instead they will show up as extension methods. This gives a way to import extension methods from just a single class, something which wasn't possible before.*

c# visual-studio-2015 static using c#-6.0

edited Aug 9 '18 at 11:49

asked Aug 6 '15 at 9:46

Cerbrus
52.5k 13 99 122

2 Answers

165

It appears the syntax has slightly changed since those blog posts were written. As the error message suggests, add `static` to your include statement:

```
using static System.Console;
//      ^
class Program
{
    static void Main()
    {
        WriteLine("Hello world!");
        WriteLine("Another message");
    }
}
```

Then, your code will compile.

Note that, in C# 6.0, this will only work for members declared as `static` .

For example, consider `System.Math` :

```
public static class Math {
    public const double PI = 3.1415926535897931;
    public static double Abs(double value);
    // <more stuff>
}
```

When `using static System.Math` , you can just use `Abs();` .
However, you'd still have to prefix `PI` because it isn't a static member: `Math.PI;` .

In C# 7.2 (and maybe lower), this shouldn't be the case, `const` values like `PI` can be used as well.

edited Aug 9 '18 at 11:34

answered Aug 6 '15 at 9:46
Cerbrus
52.5k 13 99 122

3 I'm having a similar problem, but instead with `System.Math` , specifically `Math.PI` . When I use `using static System.Math` , methods like `Sqrt()` work but not a constant like `PI` . I have to continue writing `Math.PI` or the code doesn't compile. I'm curious as to why this doesn't work. Should I submit a new question for this? – skwear Sep 22 '16 at 20:05

5 @skwear: that sounds like material for a new question, yea. – Cerbrus Sep 22 '16 at 20:41

1 As a quick answer to @skwear's question: *"using static is a new kind of using clause that lets you import **static** members <...>"*. `PI` is not a *static* member of the `Math` class. It's a constant. Simply put: the member has to be declared as `static` . – Cerbrus Apr 5 '17 at 9:54

3 Sounds like an opportunity for a new syntax like: `using const System.Math;` :) – Filip Skakun Nov 17 '17 at 0:05

1 @Cerbrus Thanks for the detail. Looks like they later noticed the inconvenience and contradiction:) – Guney Ozsan Aug 9 '18 at 11:32

0

The `static` Keyword on a `using` statement will import only the one, specified type (and it's nested types). Furthermore you must not give the type name anymore. So just add static to your using.

Note: Please use this feature only when the two classes are logically closely related, otherwise it makes reading the code pretty hard.

edited Aug 9 '18 at 11:51

answered Aug 9 '18 at 11:43
Tobias Brohl

276 5 22

"you must not give the type name" What do you mean by that? Where did I do that? – Cerbrus Aug 9 '18 at 11:45

Thobias, I'm not combining the two anywhere. – Cerbrus Aug 9 '18 at 11:49

@Cerbrus Oh the context of the 'as you did' accidentally changed, will fix that – Tobias Brohl Aug 9 '18 at 11:51