C# method call with parameter name and colon

Asked 8 years, 6 months ago Active 4 years, 1 month ago Viewed 23k times

31

5

I've begun to notice at times when I'm making method calls in C# that the names of the parameters for the method I'm calling will show up in the intellisense list appended with a colon, and that I can then format the method call thusly:

```
MethodCall(parameter1:value1, parameter2:value2);
```

Is this a new language feature? It reminds me of the way you can call stored procedures in SQL and specify parameter names like so:

```
spDoSomeStuff @param1 = 1, @param2 = 'other param'
```

Is this a similar feature? If so, to what end? If not, what is it and what is it to be used for.

c# language-features method-call

asked Mar 10 '11 at 16:25

Zannjaminderson
2,878 7 31 54

6 Answers

36

It's a new feature. See here: http://msdn.microsoft.com/en-us/library/dd264739.aspx Named parameters are standard in ObjectiveC for instance. It takes some time to get used to them but they are a good thing. Only from looking you can tell what a parameter is meant for.

answered Mar 10 '11 at 16:27

Krumelur
20.6k 20 105 220

what they all are. – Zannjaminderson Mar 10 '11 at 16:35

20

Named parameters allow you explicitly set the value of arguments in a custom order independent of the signature. Method signatures are defined by the argument types, ie, Foo(int i, bool b), which will only accept arguments of type int and bool in that order. Named arguments allow you to pass b first and i second.

answered Mar 10 '11 at 16:29

Brandon Moretz
6,293 2 25 39

2 I'm familiar with the concept of a method's signature and I wondered whether this would allow you to change the order of parameters in your call. I'm curious though - what's the point of being able to change the order? – Zannjaminderson Mar 10 '11 at 16:34

1 It's really just a code "readability" preference. The method signature is static in that when it's called the arguments will be placed on the stack in the order they are defined in the method signature. The "Named Arguments" feature just allows you to rearrange the arguments to your preference. – Brandon Moretz Mar 10 '11 at 16:40

3 @Zannjaminderson To me, the real value comes when you have methods with many default values. If you only need to change some of them you can pick them out using named parameters. This way you don't have to restate all the default values preceding the one you want to change. – André C. Andersen May 26 '13 at 9:55

@AndréChristofferAndersen, great point. That sounds useful. – Zannjaminderson May 28 '13 at 16:56

@Zannjaminderson I was actually just discussing that point as a benefit when explaining this concept to some coworkers a couple weeks ago. No idea why I didn't include it in the answer as it's definitely a huge benefit. +1 – Brandon Moretz May 28 '13 at 17:01

12

It is worth mentioning, unlike optional parameters, you can skip certain arguments and pass only the parameters you are interested in.

```
public void Example(int required, string StrVal = "default", int IntVal = 0)
{
    // ...
}

public void Test()
{
    // This gives compiler error
```

```
    Example(1, IntVal:10);
}
```

answered Mar 20 '14 at 18:11

vent
758 7 18

5

Scott Gu has introduced this new feature in his blog:

Optional Parameters and Named Arguments in C# 4

edited Dec 2 '13 at 22:38

Art
12.8k 26 75 95

answered Mar 10 '11 at 16:27

CD..
56.8k 21 119 135

2

It's the Named and Optional Parameters that came in with C# 4.

answered Mar 10 '11 at 16:27

ChrisF ♦
118k 25 224 297

0

@Krumelur said that "Named parameters are standard in ObjectiveC for instance."

That's not actually correct. Objective-C uses an infix notation, so that this message call:

```
[foo setRed:255 Green:255 Blue:0];
```

is the **setRed:Green:Blue:** message (including those colons!) with the (255,255,0) arguments interspersed within the message name.

Although, granted, at first blush Objective-C's syntax gives the appearance that Objective-C uses named parameters. But that is not actually correct, and misunderstanding the difference can be an impediment for learning Objective-C.

answered Jul 20 '15 at 18:24

John Love-Jensen

43 5