

Inline parameter name hints for C# and VB.NET in ReSharper and Rider

Posted on November 27, 2018 by Maarten Balliauw

When editing code in the latest builds of [ReSharper 2018.3 EAP](#) and [Rider 2018.3 EAP](#), the **inline parameter hints for C# and VB.NET** are something you will notice immediately. Let's see how this new feature makes code more readable, and how we can configure it.

What are inline parameter hints?

When using a literal in a method call, ReSharper and Rider will annotate those with the name of the parameter. **This makes it much easier to understand what their value represents.**

For example, we can see that the string literal "\\s" here, represents the pattern of a regular expression:

```
public IEnumerable<string> SplitWords(string sentence)
{
    if (string.IsNullOrEmpty(sentence))
    {
        return Enumerable.Empty<string>();
    }

    return Regex.Split(sentence, pattern: "\\s");
}
```

When hovering an inline parameter hint in ReSharper, we'll also display XML documentation when available:

```
var people = peopleService.Find(firstName: "Maarten", lastName: null, caseSensitive: false);
```

(parameter) bool caseSensitive
True to search case sensitive; False otherwise.

One place where inline parameter hints become very useful, is when passing in `null` values or booleans. What do the `null` and `false` arguments represent here?

```
var people = peopleService.Find("Maarten", null, false);
```

The meaning of these becomes clear immediately when using inline parameter hints, without having to rely on using [named arguments](#) for readability:

```
var people = peopleService.Find(firstName: "Maarten", lastName: null, caseSensitive: false);
```

When are inline parameter hints shown?

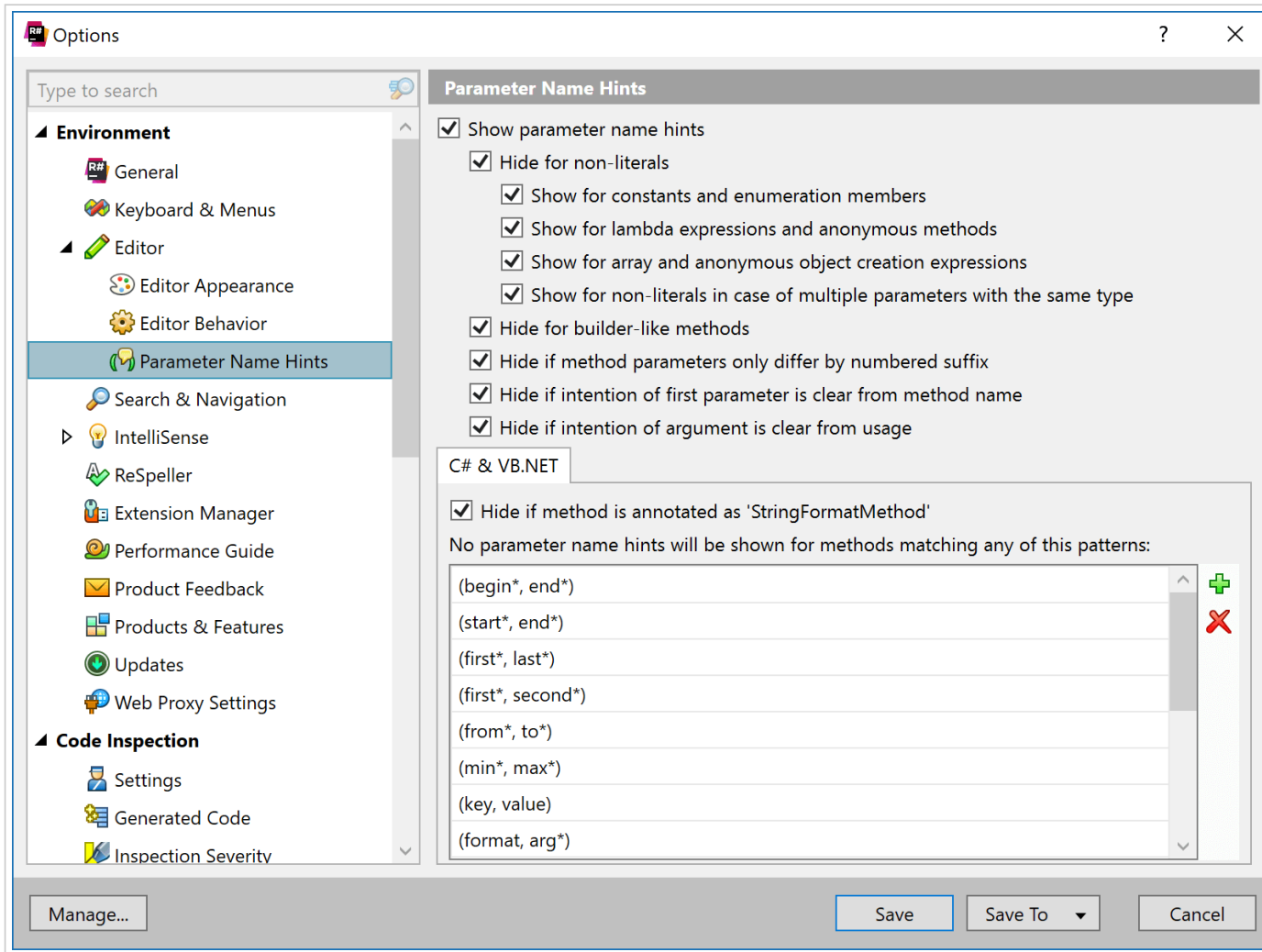
Generally speaking, ReSharper and Rider will display inline parameter hints for literals and `null` values, lambda and array expressions, anonymous methods and object creation expressions, constants and enumeration values.

ReSharper and Rider do not display inline parameter hints for *all* literals and `null` values, however. Many base class library (BCL) methods have an obvious name and functionality, and displaying inline parameter hints would not improve readability for these.

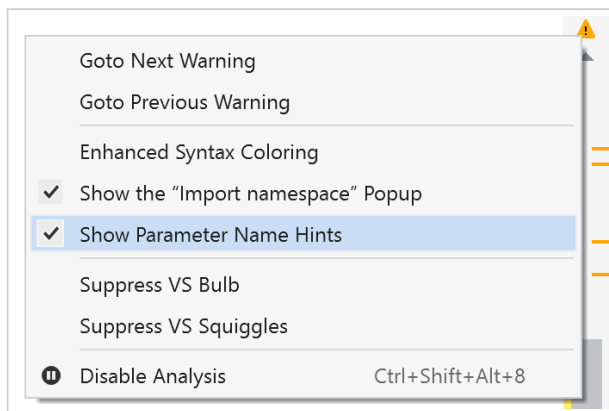
For example, in the following cases, the meaning of all parameters becomes clear from looking at their usage:

```
var range = Enumerable.Range(10, 20);  
var minimum = Math.Min(1, 2);  
  
var person1 = new Person();  
var person2 = new Person();  
var areEqual = person1.Equals(person2);  
  
var stringBuilder = new StringBuilder();  
stringBuilder.Append("Hello, ");  
stringBuilder.Append("World!");
```

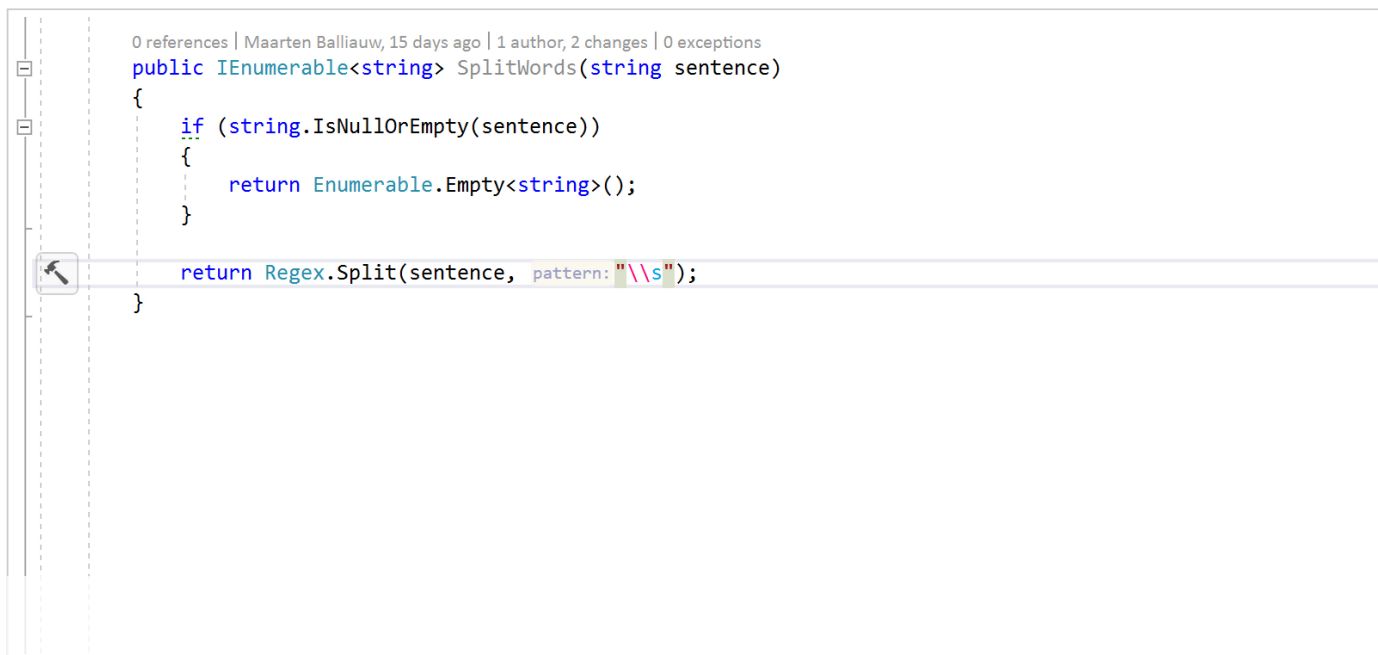
Showing (or not showing) inline parameter hints can be configured in ReSharper's settings, under **Environment | Editor | Parameter Name Hints**. Similarly in Rider, we can configure these under **Editor | Parameter Name Hints | C# and Visual Basic .NET**.



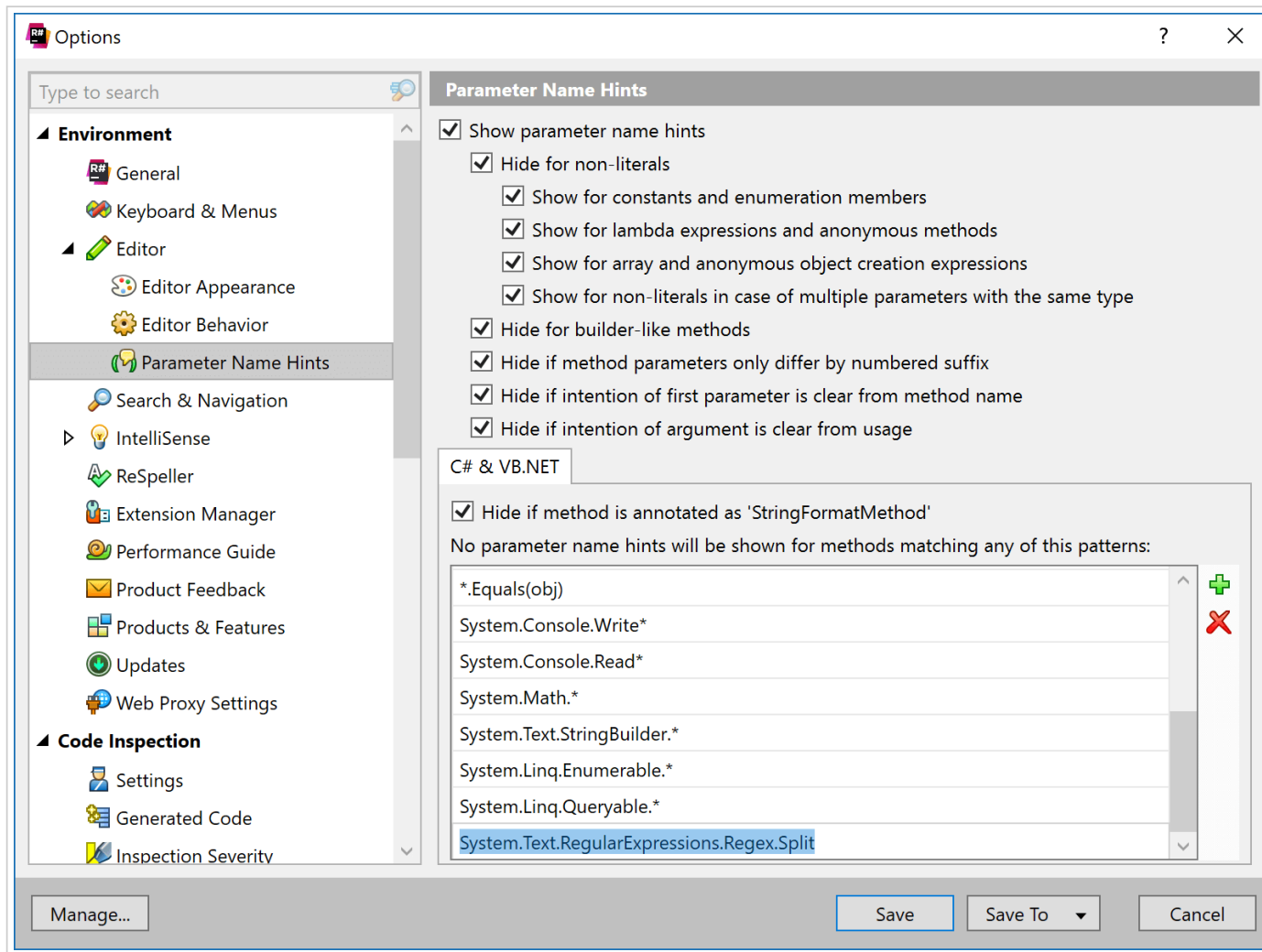
Disable inline parameter hints can be done from the settings as well. A quick way to enable/disable parameter hints is by using the status indicator context menu:



In some cases, it may be useful to disable inline parameter hints for a certain method call. This can be done using the **Configure Parameter Name Hints** action (Alt+Enter):



Once applied, the method will be added to the exclude list in the settings:



If we want to re-enable parameter hints for the `Regex.Split()` method, we can drop the entry from this list. We could also expand the entry here, and add additional namespaces and methods where inline parameter hints should be disabled.

Download [ReSharper 2018.3 EAP](#) or [Rider 2018.3 EAP](#) and give it a try! We'd love to hear your feedback!

This entry was posted in How-To's and tagged .NET Core, ASP.NET, C#, editor, ReSharper C++, Rider, VB.NET, Visual Studio. Bookmark the permalink.

26 Responses to *Inline parameter name hints for C# and VB.NET in ReSharper and Rider*



ScOtTy says:

November 27, 2018 at 2:54 pm

Sounds like a great feature!

Currently we use named parameter when calling a constructors.

Could an option be added to show parameter names (or hints) regardless of the type of the argument but based on the type of call, like constructor or method?

Reply



Maarten Balliauw says:

November 28, 2018 at 6:26 am

Thanks! Would you mind logging a feature request for this at <https://youtrack.jetbrains.com/issues/RIDER#newIssue> ?

Reply



ScOtTy says:

November 28, 2018 at 10:00 am

<https://youtrack.jetbrains.com/issue/RSRP-472497>

Done! For R# that is 😊

Reply



Maarten Balliauw says:

November 28, 2018 at 10:20 am

Thanks!

Reply

Pingback: [Dew Drop - November 28, 2018 \(#2848\) - Morning Dew](#)

**Varun Sharma** says:

December 18, 2018 at 9:06 pm

This is a great feature, Kudos to the person/team who came up with this idea. Nice.

Reply

**Jan Duchacek** says:

December 19, 2018 at 10:41 am

Hello,

I found this very distractive, so I wanted to turn it off. However in Visual Studio 2018 Professional this option is not available. In Options->Editor I have only Parameter Name Hints which does not have affect on Inline hits.

Please point me to right direction how to turn it off.

Thank you

Reply

**Maarten Balliauw** says:

December 19, 2018 at 10:48 am

Right-click a hint, then hide everywhere should do the trick

Reply

**Jan Duchacek** says:

December 20, 2018 at 5:48 am

Right-click did the trick. After that even Parameter Name Hints started to work. However you should fix the post since no Inline Hints exists in VS ReSharper options. It is very confusing.

Thank you

Reply

**Maarten Balliauw** says:

December 20, 2018 at 8:55 am

The text was OK, now have updated the screenshots as well.

Reply

BP

Ben Page says:

December 19, 2018 at 9:03 pm

These look pretty terrible on a dark theme.

Reply



Markus Janderot says:

December 20, 2018 at 10:12 am

Changing the background in Tools -> Options -> Environment -> Fonts and Colors -> ReSharper Parameter Name Hint makes them more subtle, and blend into the background.

Reply



Filiass says:

January 7, 2019 at 11:05 am

I found this. But what I didn't found is text color in tooltip window. It is some dark blue 😞

Reply



Konrad says:

December 27, 2018 at 10:07 am

This is a great feature but now text-selecting parameter feels weird. When I try to select parameter from right to left and stop before hint then first character from parameter is not selected. I need to go with cursor to the middle of parameter name to select first letter.

Also double clicking on parameter name could text-select whole parameter so it could be changed or copied.

Reply



Andrew Vasilyev says:

December 28, 2018 at 11:05 pm

Hello, Konrad! Thanks for feedback.

This is a known issue (<https://youtrack.jetbrains.com/issue/RSRP-472756>).

It is already fixed and will be out in 2018.3.2 (mid-January).

Reply



Will says:

January 4, 2019 at 1:34 pm

Thank god I'm not the only one who's struggling with this. Right now I've disabled this feature because it's insanely frustrating trying to text-select values with a parameter name hint on the left. You basically can't do it with the mouse at all as the selection stops at the character before the hint (usually a double-quote if it's a string literal).

I don't even think it's a particularly useful feature but it won't be getting enabled again until they sort this bug out (not sure how it made it out of testing like that, it cripples usability)

Reply



Matt H says:

January 8, 2019 at 10:35 am

These are great, if you find the right set of options.... Where they don't work so great are inside of .cshtml (Razor) files. Making changes to a single line causes all calls to @Html.Helper methods to re-synchronize. They "flash"-removing all hints for a few seconds and then adding them back in, which causes the display to sort of jump around. It gets pretty annoying, especially if there are a lot of @Html calls. Is there a way to turn off this feature by a specific file type?

Reply



Maarten Balliauw says:

January 8, 2019 at 10:49 am

You could try disabling them for the HtmlHelper class entirely? (blacklist much like the Regex example from this post)

Reply



E says:

January 8, 2019 at 3:11 pm

Interesting feature, but I am finding that the hints are killing VS 2018's performance.

Reply



SharpCoder says:

January 15, 2019 at 2:52 pm

Horrible feature... just makes the code harder to read. Turned off immediately.

Reply



artyom says:

February 4, 2019 at 1:49 am

Fully agree. And I spent 15min to turn it off.

Reply



Maarten Balliauw says:

February 4, 2019 at 8:57 am

Right-click a hint, then "Turn off everywhere" should do the trick.

Reply



Anang Satria says:

February 1, 2019 at 8:34 am

While it helps us to identify what is the parameter being passed but this feature makes my short method call looks very long and ugly. 😞

Reply



Maarten Balliauw says:

February 4, 2019 at 9:10 am

You can right-click them and disable hints for that specific call.

Reply



Tim Van Holder says:

February 1, 2019 at 11:47 am

Is there a way to turn them off for a particular block of code (like `//@formatter: on/off` for formatting)?

In general they are nice, but when you have a large switch statement, with single-line cases, these hints kill any attempt at having nicely aligned, readable code.

Also: configuring this in Rider (adding an exclusion pattern) does not put anything in the solution's DotSettings file, despite the "For current solution" indicator on the settings pane. Not even when explicitly saving to the team-shared solution settings.

Reply



Maarten Balliauw says:

February 4, 2019 at 9:08 am

Thanks! Logged that last one here: <https://youtrack.jetbrains.com/issue/RIDER-24154>

Would you mind creating a feature request at <https://youtrack.jetbrains.com/issues/RIDER#newIssue> for the block enable/disable?

Reply

.NET Tools Blog

Proudly powered by WordPress.