

# Code snippets for methods in Visual Studio

Asked 10 years, 10 months ago   Active 1 month ago   Viewed 51k times



In Visual Studio I can type e.g.

54

for

TAB

TAB



and a code snippet pops in.



Are there built-in code snippets for private, public, etc. methods as well?

18

visual-studio

code-snippets

edited Jul 25 at 13:09



Peter Mortensen

14.4k 19 88 117

asked Nov 15 '08 at 4:45



Edward Tanguay

83.1k 275 657 992

## 8 Answers



*ctor*: Default constructor

97

*prop*: Property



*propg*: Read-only property



*sim*: static int main method

*svm*: static void main method

There's a good list [here](#). And if you want to make your own, the [Snippet Designer](#) is very good.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

edited Jul 25 at 11:03



Peter Mortensen

14.4k 19 88 117

answered Nov 15 '08 at 4:57



Cameron MacFarland

55.2k 18 96 125

You insert it with a shortcut keyword, like `meth<Tab><Tab>`, by going into the Properties pane and setting the Shortcut option (e.g., "meth")? – [bambams](#) Sep 19 '11 at 17:10

The link to the list gives a 404 – [felixmm](#) Jan 25 '13 at 18:58

1 @FelixMM Odd it redirected for me. Anyway I've updated it to the new location. – [Cameron MacFarland](#) Jan 25 '13 at 23:38

@CameronMacFarland it's awesome. please update it with 2017 link :) – [Anirudha Gupta](#) Apr 12 '17 at 8:06

Is there no built in snippet that allows you to do something like `sim` or `svm` and allow you to tab through it and modify it? – [Samir](#) Jun 23 '17 at 9:29

## You can download the method snippets as a [Visual Studio Extension](#).

24

It supports the following:

`method` (typical method)

`vmethod` (virtual method)

`smethod` (static method)

`xmethod` (extension method)

1. In Visual Studio, go to menu *Tools* → *Extensions and Updates...*
2. Observe the *Extensions and Updates* window
3. Enter "C# Methods Code Snippets" in the search field (upper right)

edited Jul 25 at 12:05



Peter Mortensen

14.4k 19 88 117

answered Apr 23 '15 at 13:22





Scott Nimrod

5,991 3 38 70

This is the only answer that actually answers the question - code snippets for **methods** (from the question: "...code snippets for private, public, etc.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

looked at the first 20 pages of 25 hits each (no end in sight). But I did find a number interesting items: "*Import Code Snippet*", "*CodeHelperTools*", "*NUnit VS Templates*", "*StyleCop*", "*Language Convert*" (C# <-> VB.NET), "*Script#*" (transpile from C# to JavaScript), – [Peter Mortensen](#) Jul 25 at 12:57 

cont' - "*Refactor*" (.NET assembly browser and decompiler), "*kiss!*" (template-based code generator for C#, VB.NET, and SQL), "*WPF MVVM project template*", "*T4Awesome*", "*SharpSetup*" (WiX, create installers), "*Units of Measurements*", "*.NET FTP library*", and "*Comment*". – [Peter Mortensen](#) Jul 25 at 12:59 

Below are the steps I used to create a custom snippet for Visual Studio 2010, but the steps work for Visual Studio 2008.

12

Create a new text file named *method.snippet* and paste the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<CodeSnippets xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
  <CodeSnippet Format="1.0.0">
    <Header>
      <Title>method</Title>
      <Shortcut>method</Shortcut>
      <Description>Code snippet for method</Description>
      <Author>Kevin Hogg</Author>
      <SnippetTypes>
        <SnippetType>Expansion</SnippetType>
      </SnippetTypes>
    </Header>
    <Snippet>
      <Declarations>
        <Literal>
          <ID>methodname</ID>
          <ToolTip>Method name</ToolTip>
          <Function>MethodName()</Function>
          <Default>MethodNamePlaceholder</Default>
        </Literal>
      </Declarations>
      <Code Language="csharp"><![CDATA[public void $methodname$ ()
{
    $end$
}]]>
    </Code>
  </Snippet>
</CodeSnippet>
</CodeSnippets>
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

- Visual Studio 2010: C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC#\Snippets\1033\Visual C#
- Visual Studio 2008: C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC#\Snippets\1033\Visual C#

Once you save your file, the snippets are automatically loaded, so you can now open Visual Studio and type:

```
method<tab><tab>
```

\*where <tab> is the  key on your keyboard.

You should now see the following created, with the `MethodNamePlaceholder` highlighted so you can change the name.

```
public void MethodNamePlaceholder()  
{  
  
}
```

edited Jul 25 at 11:39



[Peter Mortensen](#)

14.4k 19 88 117

answered Oct 8 '13 at 12:32




[Kevin Hogg](#)

1,612 21 28

---

Please note I could only test this in Visual Studio 2010 – [Kevin Hogg](#) Oct 8 '13 at 12:33

---

1 It also works on VS2013 Path : C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC#\Snippets\1033\Visual C# – [zapoo](#) Aug 4 '15 at 14:21 

---

Works VS2015. Thanks. Great DYI. Check the box for learned something new today. – [Rourke](#) Sep 22 '17 at 13:38

---

▲ If you want to see the list of all available snippet :

9

- Press  +  and then  + 

edited Feb 18 at 0:37

answered Mar 14 '17 at 5:29



[ANewGuyInTown](#)

2,359 1 15 18

---

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

That's right. It depends on how you formulate it. What was missing was "press and hold CTRL". But this works too. – [Samir](#) Feb 26 at 22:41

Some of the snippets I use, also mentioned at MSDN, follows:

6

1. **#if** Creates a #if directive and a #endif directive.
2. **#region** Creates a #region directive and a #endregion directive.
3. **~** Creates a destructor for the containing class.
4. **attribute** Creates a declaration for a class that derives from Attribute.
5. **checked** Creates a checked block.
6. **class** Creates a class declaration.
7. **ctor** Creates a constructor for the containing class.
8. **cw** Creates a call to WriteLine.
9. **do** Creates a do while loop.
10. **else** Creates an else block.
11. **enum** Creates an enum declaration.
12. **equals** Creates a method declaration that overrides the Equals method defined in the Object class.
13. **exception** Creates a declaration for a class that derives from an exception (Exception by default).
14. **for** Creates a for loop.
15. **foreach** Creates a foreach loop.
16. **forr** Creates a for loop that decrements the loop variable after each iteration.
17. **if** Creates an if block.
18. **indexer** Creates an indexer declaration.
19. **interface** Creates an interface declaration.
20. **invoke** Creates a block that safely invokes an event.
21. **iterator** Creates an iterator.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

24. **mbox** Creates a call to `MessageBox.Show`. You may have to add a reference to `System.Windows.Forms.dll`.
25. **namespace** Creates a namespace declaration.
26. **prop** Creates an auto-implemented property declaration.
27. **propfull** Creates a property declaration with get and set accessors.
28. **propg** Creates a read-only auto-implemented property with a private "set" accessor.
29. **sim** Creates a static int Main method declaration.
30. **struct** Creates a struct declaration.
31. **svm** Creates a static void Main method declaration.
32. **switch** Creates a switch block.
33. **try** Creates a try-catch block.
34. **tryf** Creates a try-finally block.
35. **unchecked** Creates an unchecked block.
36. **unsafe** Creates an unsafe block.
37. **using** Creates a using directive.
38. **while** Creates a while loop.

edited Jul 25 at 11:43



Peter Mortensen

14.4k 19 88 117

answered Sep 22 '16 at 4:48



UJS

680 1 7 16



1

I made my own snippet for a method. The XML code for it is the following, and you can add it to a file called "my\_method.snippet" (or whatever\_you\_want.snippet) in `C:\Users\YOUR_USERNAME\Documents\Visual Studio 2012\Code Snippets\Visual C#\My Code Snippets` (your path might be different because I use VS2012):



```
<CodeSnippet Format="1.0.0">
  <Header>
    <Title>method</Title>
    <Shortcut>method</Shortcut>
    <SnippetTypes>
      <SnippetType>Expression</SnippetType>
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```

<Snippet>
  <Declarations>
    <Literal>
      <ID>access_modifier</ID>
      <Default>private</Default>
    </Literal>
    <Literal>
      <ID>return_type</ID>
      <Default>void</Default>
    </Literal>
    <Literal>
      <ID>name</ID>
      <Default>New_method</Default>
    </Literal>
  </Declarations>
  <Code Language="csharp">
    <![CDATA[ $access_modifier$ $return_type$ $name$ ()
{
$end$
}]]>
  </Code>
</Snippet>
</CodeSnippet>

```

answered May 12 '13 at 17:35

**daniel1426****127** 1 13

0



You can create customs snippets. Like this:

<http://www.mediafire.com/file/gz3tzjnydk5/meth.snippet>

answered Mar 26 '10 at 23:34

**EdgarT****740** 8 16

That link is for a page about the meth.snippet XML document format (.SNIPP). That is far away from being about custom snippets. I suggest adding an example and/or more information to make this answer more useful. – [Peter Mortensen](#) Jul 25 at 11:35

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



The code snippet for properties is:

-1



prop

edited Feb 14 '18 at 9:13



Mong Zhu

17.4k 7 28 55

answered Nov 15 '08 at 4:48



Soraz

3,379 3 25 45

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).