kostadin@gmail.com ▾

Edit this form

55 responses

View all responses Publish analytics

Summary

How many years of experience in using the Visual Studio IDE do you have?

15

10

4

5

1

2

8

6

9

7

11

30

14

0

13

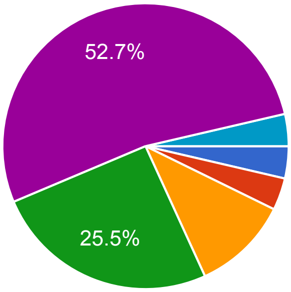
18

12

3

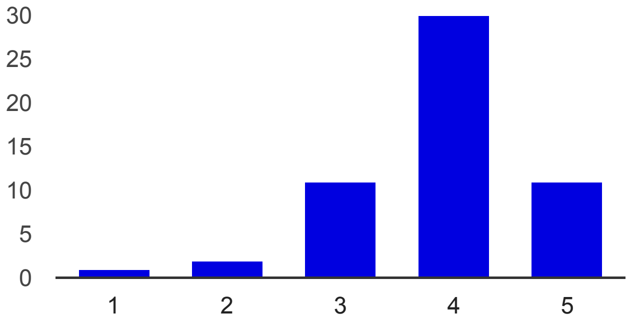
25

Which version of Visual Studio do you use most of the time?



Visual Studio 2012	**14**	25.5%
Visual Studio 2013	**29**	52.7%
Visual Studio 2015	**2**	3.6%

How would you rate the usability (ease of use) of the Visual Studio IDE?

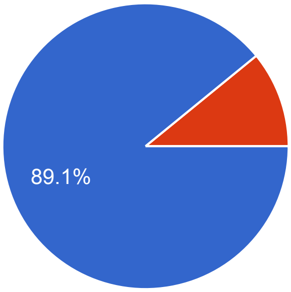


Not Usable: 1	**1**	1.8%
2	**2**	3.6%
3	**11**	20%
4	**30**	54.5%
Extremely Usable: 5	**11**	20%

'Find in Files' Search in Visual Studio

[Image]

Have you ever performed a 'Find in Files' search using Visual Studio?

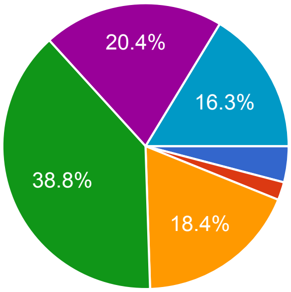


Yes	**49**	89.1%
No	**6**	10.9%

'Find in Files' Search in Visual Studio

[Image]

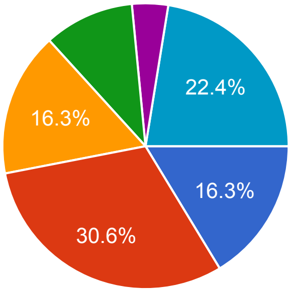
How often do you use 'Find in Files' to search all of the code in the currently open Visual Studio solution?



Never	**2**	4.1%
About once per month	**1**	2%
About once per week	**9**	18.4%
About once per day	**19**	38.8%
About once per hour	**10**	20.4%
More frequently then once per hour	**8**	16.3%

On average, how many search results do you need to examine before locating a useful program element?

1	**8**	16.3%
2	**15**	30.6%
3	**8**	16.3%
4	**5**	10.2%



5	**2**	4.1%
More than 5	**11**	22.4%

e way 'Find in Files' search works

No

The output is too cluttered. Results could be bit more organized in terms of folder/solution/projects needs to be better than notepad++ for an IDE

It is difficult to find anything in Find Results since the results and their line numbers are not put nicely to columns. The results look like a mess.

Auto filling with signs which were in cursor line.

It's difficult to find specific elements and navigate in hierarchy. It's like poor man's search feature. On the other hand it provides a clear and concise list of usages of something if you know when it's the right time to use it. Usually I use Resharper navigation features, "Find usages" and inspection features, but sometimes "find in files" is still useful.

It is not as intuitive as other tools I have been using

no

In most cases I use it for finding in files not in a visual studio solution. For such searches it is usually a little bit harder to fid the right finding.

Search is disconnected from program language context (object type, interface implemented...)

It's differecult to return back

The search function often searches binary files

The keyboard shortcuts Ctrl+F and Shift+Ctrl+F work in different way in different VisualStudio versions (this is very inconvenient, because I must daily use several different VisualStudio versions).

Not that I can think of at the moment.

no difficulties

No difficults.

I used it years ago. Not used it recently

Sometimes forget to check the setting for solution only or solution + dependancies

I wish it was faster

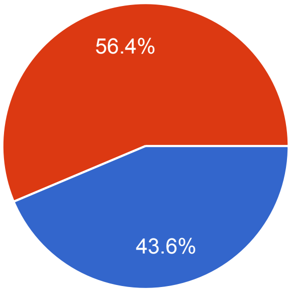
Find is not precise. less advanced options for element search

filename and path takes up most of the search result line, so I'm always scrolling to the right to see what the matching line says. no context for the method enclosing the matching line.

Working Set Support in Visual Studio

[Image]

Have you ever navigated between active files using the Ctrl+Tab or Ctrl+Shift+Tab keystrokes in Visual Studio?

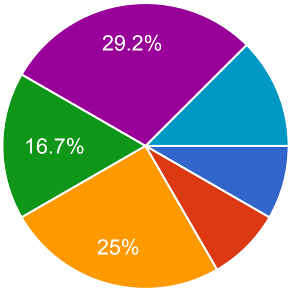


Yes	**24**	43.6%
No	**31**	56.4%

Working Set Support in Visual Studio

[Image]

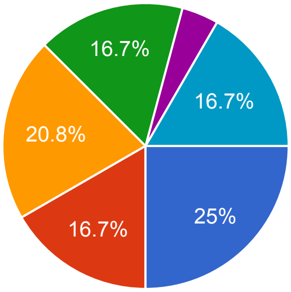
How often do you use the active file navigation capability, invoked using the Ctrl+Tab or Ctrl+Shift+Tab keystrokes, in Visual Studio?



Never	**2**	8.3%
About once per month	**2**	8.3%
About once per week	**6**	25%
About once per day	**4**	16.7%

About once per hour	**7**	29.2%
More frequently than once per hour	**3**	12.5%

On average, how many files do you navigate to before opening the file you are looking for?



1	**6**	25%
2	**4**	16.7%
3	**5**	20.8%
4	**4**	16.7%
5	**1**	4.2%
More than 5	**4**	16.7%

Are there any difficulties you have faced using the the active file navigation capability in Visual Studio?

The list is too long sometimes.

no

Nope

We use Resharper Ctrl+T

no difficulty

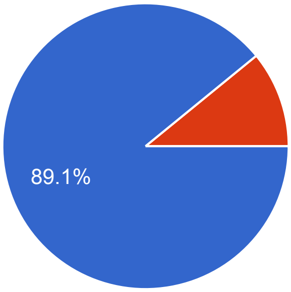
Rarely use it. I usually know what object I need and use the tab to go directly there.

Debugging Support for Stepping Through Code in Visual Studio

[Image]

Have you ever used any of the stepping commands (i.e. StepInto, StepOver, RunToCursor) to debug your code?

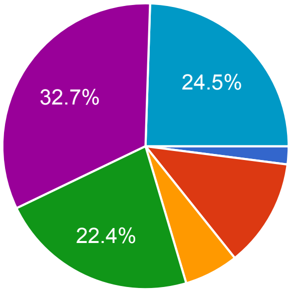
Yes	**49**	89.1%
No	**6**	10.9%



rough Code in Visual

[Image]

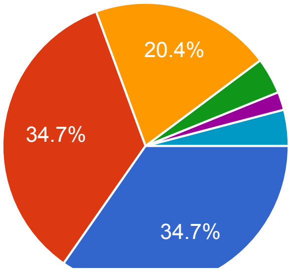
How often do you use the StepInto or StepOver command to debug in Visual Studio?



Never	**1**	2%
About once per month	**6**	12.2%
About once per week	**3**	6.1%
About once per day	**11**	22.4%
About once per hour	**16**	32.7%
More frequently than once per hour	**12**	24.5%

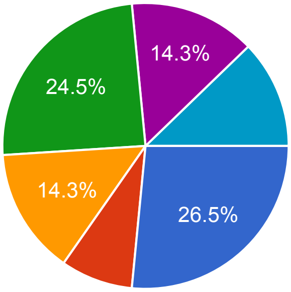
Consider a scenario where you have stopped on a breakpoint using the debugger. On average, how many StepOver or StepInto commands do you execute in one sequence before you resume execution or end the debugging session?

0-10	**17**	34.7%
10-20	**17**	34.7%
20-30	**10**	20.4%
30-40	**2**	4.1%
40-50	**1**	2%



More than 50	2	4.1%

a breakpoint using the
or command to execute multiple



Never	13	26.5%
About once per month	4	8.2%
About once per week	7	14.3%
About once per day	12	24.5%
About once per hour	7	14.3%
More frequently than once per hour	6	12.2%

Describe a scenario where you debugged a longer set of statements. How do you use the StepInto, StepOver and RunToCursor commands to perform that activity?

Use Stepinto when ever we need to step into a specific function in a longer set of statements, stepover if the function is not important and RunToCursor to skip looping statements

The questions are more related to a specific task. Each task can have different scenarios, so this survey is not giving the full picture...

I try to set breakpoints so that I need to step as little as possible. I use RunToCursor when there is more than ~10 code lines to step over.

-

no

Break to the beginning of interesting segment. Hitting F10 to the end and checking values between. I use new break point and F5 rather than RunToCursor. I am working on a real-time software where most of the code runs every cycle and the cycles are short. The break point is there for the next cycle.

I usually only do StepOver, and activate a number of Watches to see what happens with my state

For complicated debug scenarios I do not consider these commands as of much use

Set a break point at a particular If condition, step into it to see which path it takes, use RunToCursor to get it to the next spot in which I want to check a value.

I only use StepInto and StepOver. Depends on how integrated the code is between files.

Answers are dependent of the goal of the task. If you are doing a new functionality, then you do all these operations less than if you are fixing a bug

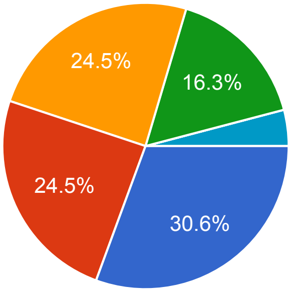
I set a breakpoint at the call of a function I want to debug. Then I use step into to debug into the function and to debug into other functions where that I want to investigate. Whenever I am sure that a functions is error-free I use step-over to execute this funtion. I try to avoid functions with lots of lines of code. So I do not use RunToCursor very often.

I use the step-commands to explore the code and what is happening. Until I get a "AHA" experience: that's why it doesn't work.

Generally I set a breakpoint in roughly the right place, step through and then adjust my breakpoints to the interesting bits. Then I just run to get to each point, rather than step.

I generally place a break point at the method call where I think the problem exists. I will then step into the method, then step over any sub methods, (checking output and reference variables from the calls). If I see anything I don't expect, I will drag the next instruction pointer back to the method and step into the method to try to figure out why. A lot of this depends on the issue I am trying to fix, and the complexity, type and size of the application.

How often do you use conditional breakpoints in Visual Studio?



Never	**15**	30.6%
About once per month	**12**	24.5%
About once per week	**12**	24.5%
About once per day	**8**	16.3%
About once per hour	**0**	0%
More frequently than once per hour	**2**	4.1%

Are there any difficulties you have faced in using conditional breakpoints to debug in Visual Studio?

Specifying the conditional expression is complex

A breakpoint can't be assigned conditions during creation, it's a two step process. I feel that a option for creating a conditional breakpoint and adding the condition directly while the breakpoint is being create would be nice.

They can be painfully slow depending on code and objects in question. One problem is that you can't set data breakpoints in C#. It's often important to know when the valúe changes.

No

no

Only difficulty has been computation speed when having a conditional break in a loop. It is much slower than writing an "if" statement into the loop and breaking there.

The debugging is slowed down so much that I have stopped using it. It would be very useful for me, since I often want to se what happens at som specific state in a longer run.

Makes debugging extremely slow. I rather update source code to put the condition within the if statement and break over there

Setting up the 'correct' condition can be sometimes bothersome and the logic as it is now is not given to complicated scenarios

Nope

no difficulty

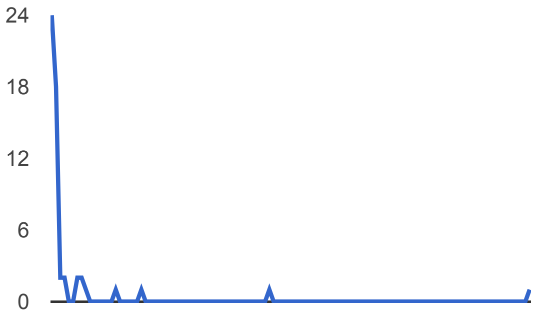
performance

I often had the problem that the condition statement could not be evaluated. Then the debugger did not stop and I hat to check the condition again.

Not enough experience

Really don't use them that often.

Number of daily responses


24
18
12
6
0