

Visual Studio, debug one of multiple threads

Asked 8 years, 9 months ago Active 3 months ago Viewed 42k times



118



16

I have an application with 4 threads working the same code. However, when I step it jumps between the different threads. How can I lock it to one thread so the other threads are ignored for debugging?

c#

multithreading

visual-studio-2010

asked Oct 10 '10 at 19:03



Oskar Kjellin

15.3k 10 47 91

which version of Visual Studio are you using ? Express, Pro, Ultimate .. ? – Mark Oct 10 '10 at 19:06

@Mark I am using ultimate – Oskar Kjellin Oct 10 '10 at 19:09

then jeffamaphone's link will help and maybe this also for further information to switch to another thread while debugging msdn.microsoft.com/en-us/library/bb157786.aspx – Mark Oct 10 '10 at 19:10

Thanks everybody :) – Oskar Kjellin Oct 10 '10 at 19:13

See the related discussion: [visualstudio.uservoice.com/forums/121579-visual-studio/...](https://visualstudio.uservoice.com/forums/121579-visual-studio/) – user626528 Jan 25 '18 at 22:45

6 Answers



101



Yes.

In the Threads window (Debug -> Windows -> Threads) right-click the thread you want and select "switch to thread".

You can also choose "freeze" on the threads you don't want to debug in order to keep them from running. Don't forget to "thaw" them if you expect them to do work, however.

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

answered Oct 10 '16 at 13:00

[i_am_jorf](#)

45.6k

12

116

209

20 I'm confused. Is the answer "It cannot be done?" The question asks how to stay locked to a particular thread so the debugger won't jump around between them. Switching to a thread is fine, but as soon as another thread does something, then debugger jumps to it. If I can't freeze the other thread because it needs to do stuff, then how do I stay locked to just the thread I'm concerned with? – [bubbleking](#) Feb 10 '16 at 16:32

Incomplete :| you have to always click on switch to, every time something happens – [deadManN](#) Jan 7 '17 at 6:20



Single stepping through a single thread seems to be mostly fixed in VS 2012 (with some caveats you can see in my link below). Breakpoints are a pain.

16



Freezing and thawing threads is the usual workaround, as previous answers have stated, but it's tedious, and it can cause hangs when your thread waits on another thread that's frozen. These can be tough to recover from without losing your place in your thread of interest.

Another useful workflow is to apply a thread filter on your breakpoints, also stated in some of the answers:

Create a breakpoint, right click on the breakpoint, click Filter, and enter ThreadId = 7740 (your thread id from the threads window).

This can be very tedious.

My suggestion to Microsoft is to fix single stepping (and variations of it) to never switch threads unless an explicit breakpoint is hit in another thread. They should also add a shortcut (maybe Ctrl-F9) to create a breakpoint with the current thread id as its filter. This would make the second workflow much more convenient.

Vote up the suggestion if you agree this would be useful, or add your own suggestions:

<https://visualstudio.uservoice.com/forums/121579-visual-studio/suggestions/8543248-make-the-debugger-stick-to-the-current-thread-inst>

answered Jul 2 '15 at 17:42

[Steve](#)

454

4

10

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



10

You could also put a conditional breakpoint in your code and put the `thread.Id == [someValue] or Thread.Name == "[Somename]"` in the breakpoint condition...

edited Feb 6 '13 at 8:44



xlecoustillier

14.1k 12 51 74

answered Oct 10 '10 at 19:14



Charles Bretana

117k 20 128 200

Thanks Charles, that was helpful (did not know you could do that). However, the most efficient way for me to debug is the one that jeffamaphone wrote as I do not know the name before it hits the breakpoint and sees some values – [Oskar Kjellin](#) Oct 10 '10 at 19:17



2

This strongly resembles a very similar problem in Visual Studio 2008 SP1. It was fixed with a post-SP hotfix. But there's other evidence that the hotfix didn't get incorporated into the code base, this [feedback item](#) was also a problem. It isn't that unusual for hotfixes to not get integrated back.

There isn't a feedback item that exactly describes your problem, at least that I can find. I'd recommend you file one. Given the usual trouble with reproduce bugs like this, I'd strongly recommend you include a reproduction project that exhibits this problem with instructions on how to reproduce the issue.

There is a workaround of sorts for your issue, you could go into Debug + Windows + Threads, right-click the threads you don't want to debug and select Freeze. Don't forget to Thaw them later.

These bugs were fixed again in Visual Studio 2010 Service Pack 1.

edited Sep 25 '12 at 18:09



Peter Mortensen

14.3k 19 88 116

answered Oct 10 '10 at 19:20



Hans Passant

809k 113 1376 2158

A much quicker workaround exists for simple cases - see comments in Steve's link.



the debugger will only ever complete a step on the thread from which the step was originated. So if you hit a breakpoint, disable it, and then begin stepping you should not stop on a different thread. If you have other breakpoints in your application and another

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

So in my case once the various threads started hitting my breakpoint I just hit continue a few times until I identified the call I was looking for - then removed the breakpoint and stepped through the rest of the code while remaining on the same thread without interference from the rest of them.

This obviously becomes a problem if you have multiple breakpoints that you want to keep, etc. - but again for simple cases this is much easier to do.

answered May 25 '17 at 16:25



Boris

810 1 15 26



1



I'm Using Visual Studio Professional 2017, and I use the Threads window to selectively freeze and thaw threads. Usually I have multiple threads of the same code and I only want to freeze them, not others. I actually like the MS Threads window because I can select a subset of threads to freeze. I group the threads by name and can then freeze all the ones running the same code as I am debugging while letting the remaining threads run. I tried using the Erwin Mayer extension, and it worked very well, but it freezes all threads except the one I am running, and I sometimes get into a situation when debugging does not hit the breakpoint I think it should, then because all of the other threads are stopped and the application seems halted. Hitting the pause button, and unfreezing threads in the threads window fixes that issue.

answered Apr 23 at 14:07



BlueMax

11 2