DEBUGGING

By Philip Sutton

WHO AM I

- Started full time work 1991
- Worked for Science, Travel, Management Consultancy, Finance, Medical, Education
- Programmed in COBOL, Fortran, Visual Basic 6, VB.net and C# as well as Sql
- C# since 2005
- Twitter @GBPhilipSutton

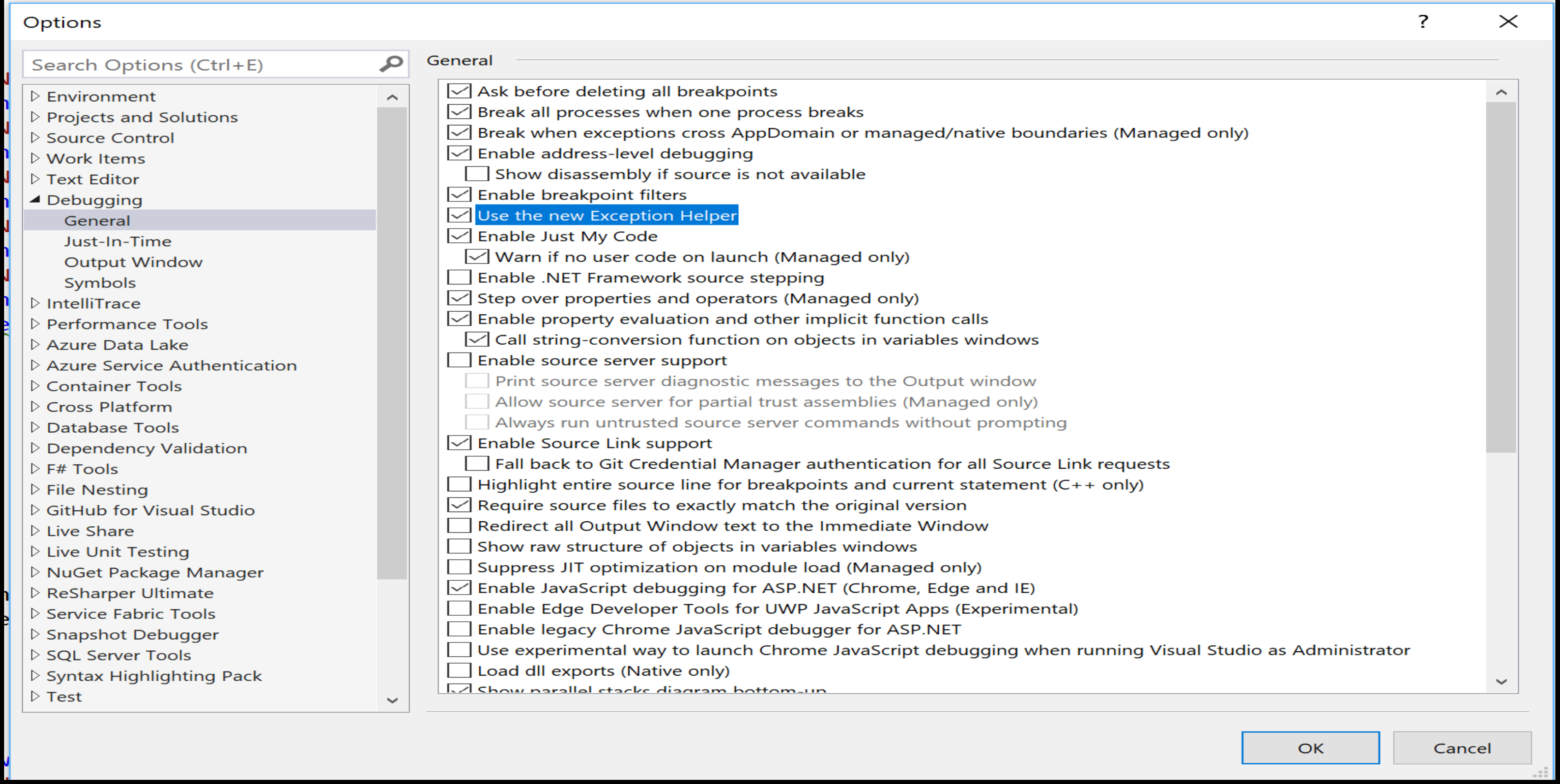
INTRODUCTION

- Visual Studio Options
- Debugging Attributes
- Datatips
- Debugging Windows
- Dumps
- Symbol Servers
- Front End
- Enterprise Features

VISUAL STUDIO OPTIONS

- Debugging Options
- Exception Options

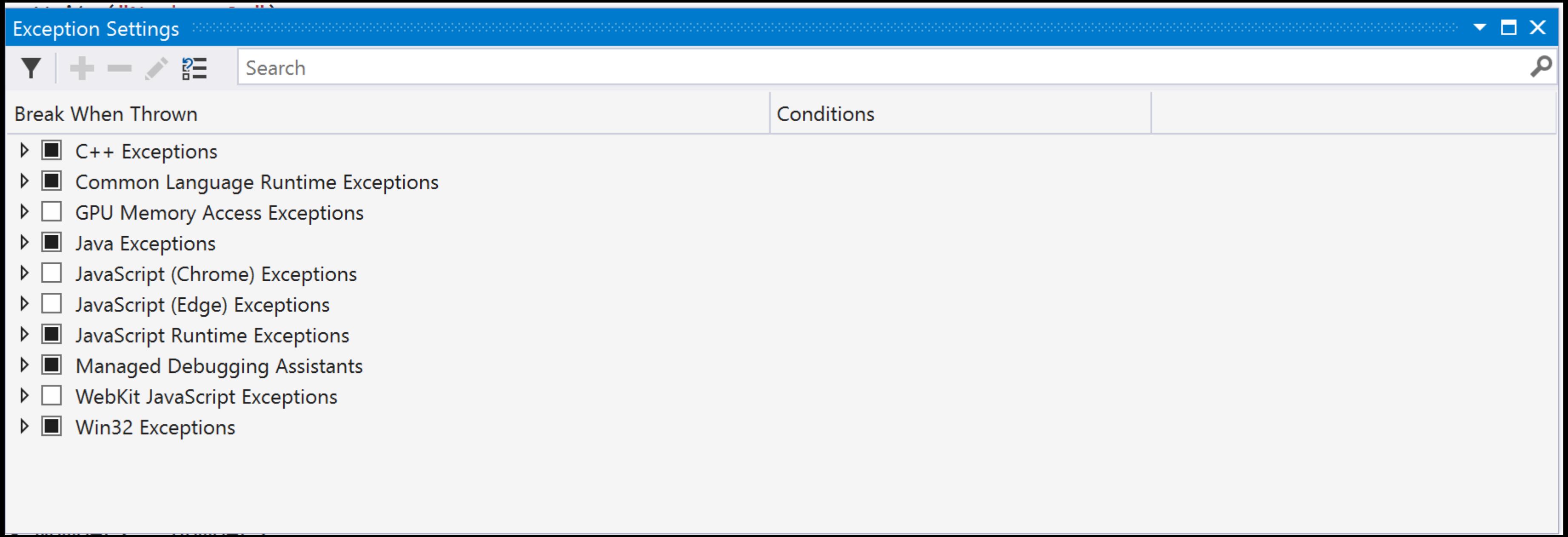
DEBUGGING OPTIONS



General

- ☑ Ask before deleting all breakpoints
- ☑ Break all processes when one process breaks
- ☑ Break when exceptions cross AppDomain or managed/native boundaries (Managed only)
- ☑ Enable address-level debugging
  - ☐ Show disassembly if source is not available
- ☑ Enable breakpoint filters
- ☑ Use the new Exception Helper
- ☑ Enable Just My Code
  - ☑ Warn if no user code on launch (Managed only)
- ☐ Enable .NET Framework source stepping
- ☑ Step over properties and operators (Managed only)
- ☑ Enable property evaluation and other implicit function calls
  - ☑ Call string-conversion function on objects in variables windows
- ☐ Enable source server support
  - ☐ Print source server diagnostic messages to the Output window
  - ☐ Allow source server for partial trust assemblies (Managed only)
  - ☐ Always run untrusted source server commands without prompting
- ☑ Enable Source Link support
  - ☐ Fall back to Git Credential Manager authentication for all Source Link requests
- ☐ Highlight entire source line for breakpoints and current statement (C++ only)
- ☑ Require source files to exactly match the original version
- ☐ Redirect all Output Window text to the Immediate Window
- ☐ Show raw structure of objects in variables windows
- ☐ Suppress JIT optimization on module load (Managed only)
- ☑ Enable JavaScript debugging for ASP.NET (Chrome, Edge and IE)
- ☐ Enable Edge Developer Tools for UWP JavaScript Apps (Experimental)
- ☐ Enable legacy Chrome JavaScript debugger for ASP.NET
- ☐ Use experimental way to launch Chrome JavaScript debugging when running Visual Studio as Administrator
- ☐ Load dll exports (Native only)
- ☑ Show parallel stacks diagram bottom-up

EXCEPTION OPTIONS



Break When Thrown | Conditions

- ■ C++ Exceptions
- ■ Common Language Runtime Exceptions
- □ GPU Memory Access Exceptions
- ■ Java Exceptions
- □ JavaScript (Chrome) Exceptions
- □ JavaScript (Edge) Exceptions
- ■ JavaScript Runtime Exceptions
- ■ Managed Debugging Assistants
- □ WebKit JavaScript Exceptions
- ■ Win32 Exceptions

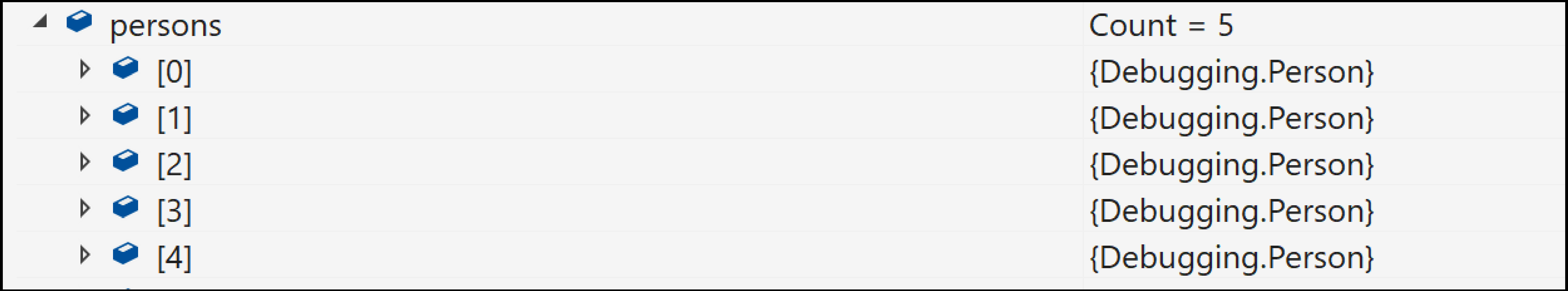
EXCEPTIONS SETTINGS

- Demo

DEBUGGING ATTRIBUTES

- Debugger Display
- Debugger Browsable
- Debugger Type Proxy

DEBUGGERDISPLAY

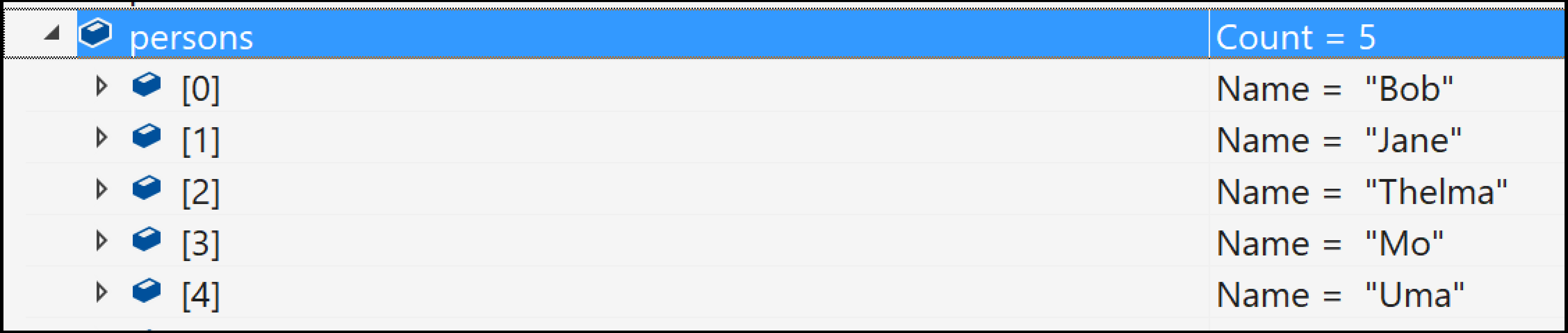


```
◢ persons                    Count = 5
   ▷ [0]                     {Debugging.Person}
   ▷ [1]                     {Debugging.Person}
   ▷ [2]                     {Debugging.Person}
   ▷ [3]                     {Debugging.Person}
   ▷ [4]                     {Debugging.Person}
```

DEBUGGERDISPLAY

```
• [DebuggerDisplay("Name =  {Name}")]
•     public class Person
•     {
•         public string Name { get; set; }
•         public int Age { get; set; }
•     }
```

DEBUGGERDISPLAY



```
◢ persons                Count = 5
  ▷ [0]                  Name =  "Bob"
  ▷ [1]                  Name =  "Jane"
  ▷ [2]                  Name =  "Thelma"
  ▷ [3]                  Name =  "Mo"
  ▷ [4]                  Name =  "Uma"
```

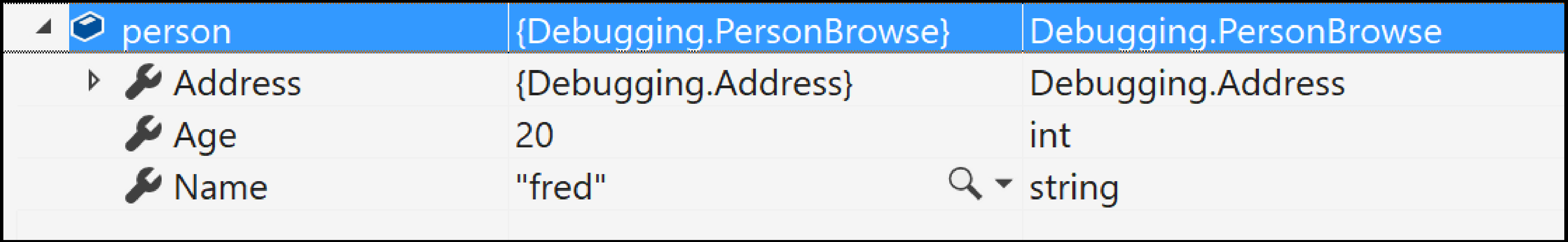
DEBUGGERBROWSABLE

```
• public class PersonBrowse
•   {
•       public string Name { get; set; }

•       public int Age { get; set; }

•       [DebuggerBrowsable(DebuggerBrowsableState.Collapsed)]
•       public Address Address { get; set; }
•   }
```

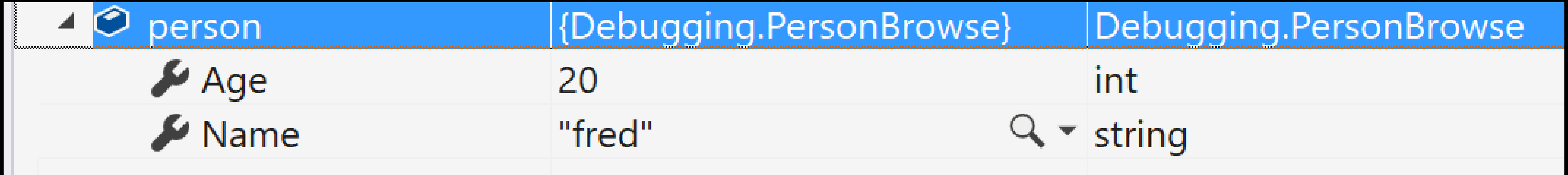
DEBUGGERBROWSABLE



person	{Debugging.PersonBrowse}	Debugging.PersonBrowse
Address	{Debugging.Address}	Debugging.Address
Age	20	int
Name	"fred"	string

DEBUGGERBROWSABLE

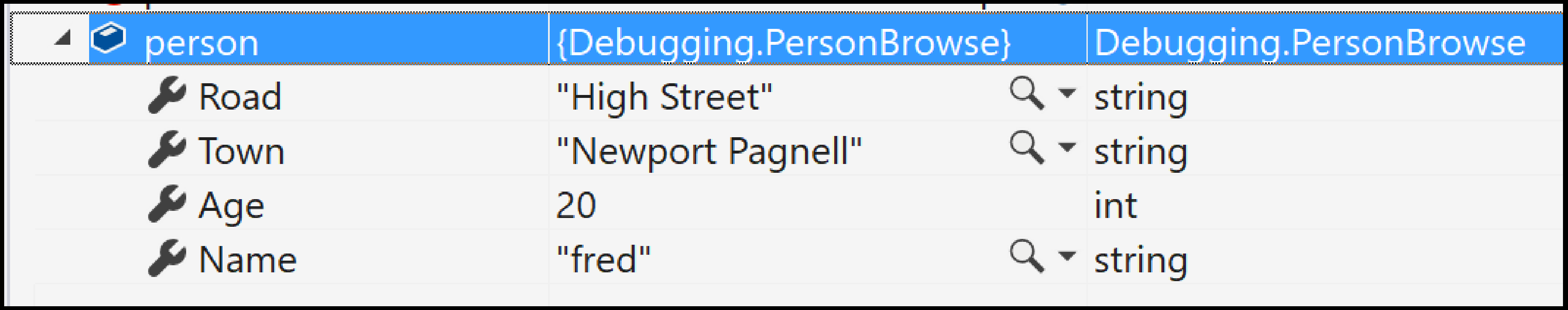
- Never



◢ person	{Debugging.PersonBrowse}	Debugging.PersonBrowse
Age	20	int
Name	"fred"	string

DEBUGGERDISPLAY

- RootHidden



◢ person	{Debugging.PersonBrowse}	Debugging.PersonBrowse
Road	"High Street"	string
Town	"Newport Pagnell"	string
Age	20	int
Name	"fred"	string

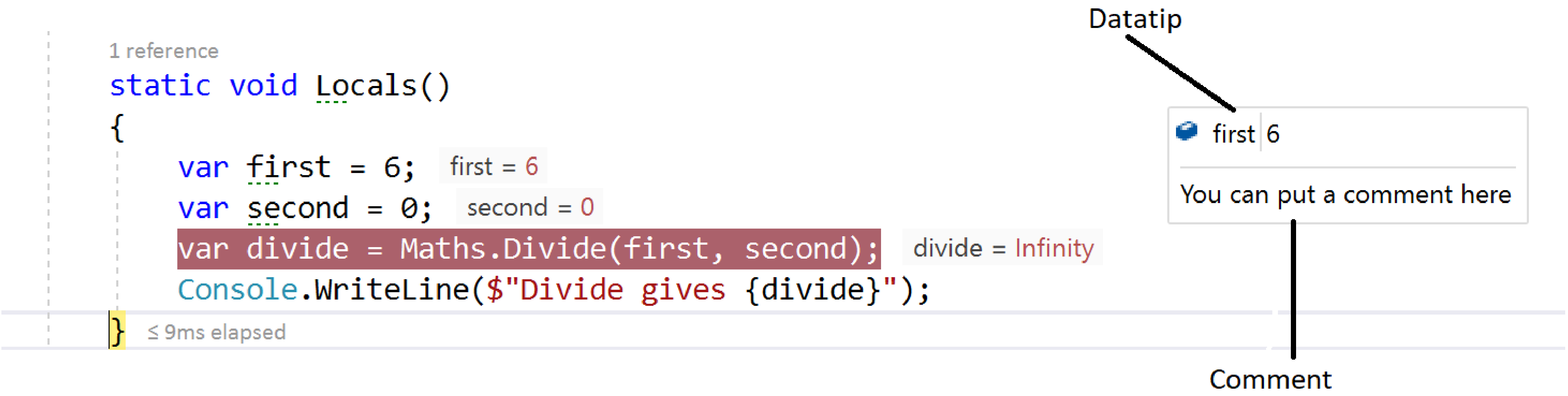
DEBUGGER PROXY TYPE

- Allows the significant and fundament change on the debugging view of a type.
- Demo

DEBUGGER VISUALISERS

- Some built in – xml, json, text, html
- Many in the market place
- Write your own

DATATIPS

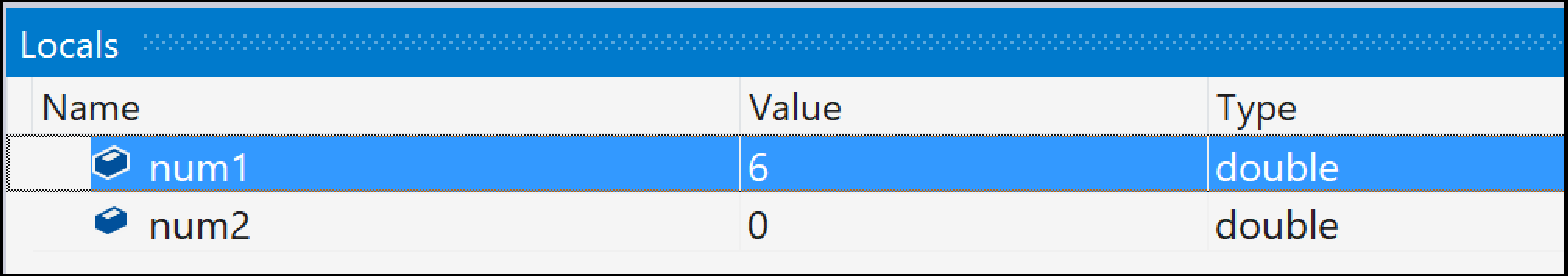


```
1 reference
static void Locals()
{
    var first = 6;  first = 6
    var second = 0;  second = 0
    var divide = Maths.Divide(first, second);  divide = Infinity
    Console.WriteLine($"Divide gives {divide}");
}  ≤ 9ms elapsed
```

first 6

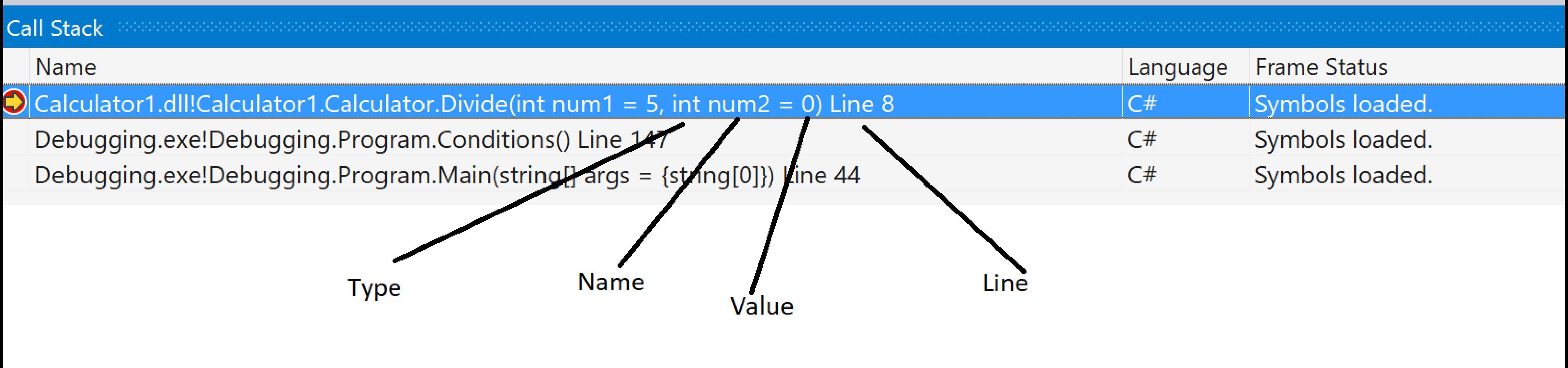
You can put a comment here

DEBUGGING WINDOWS - LOCALS/AUTO



Name	Value	Type
num1	6	double
num2	0	double

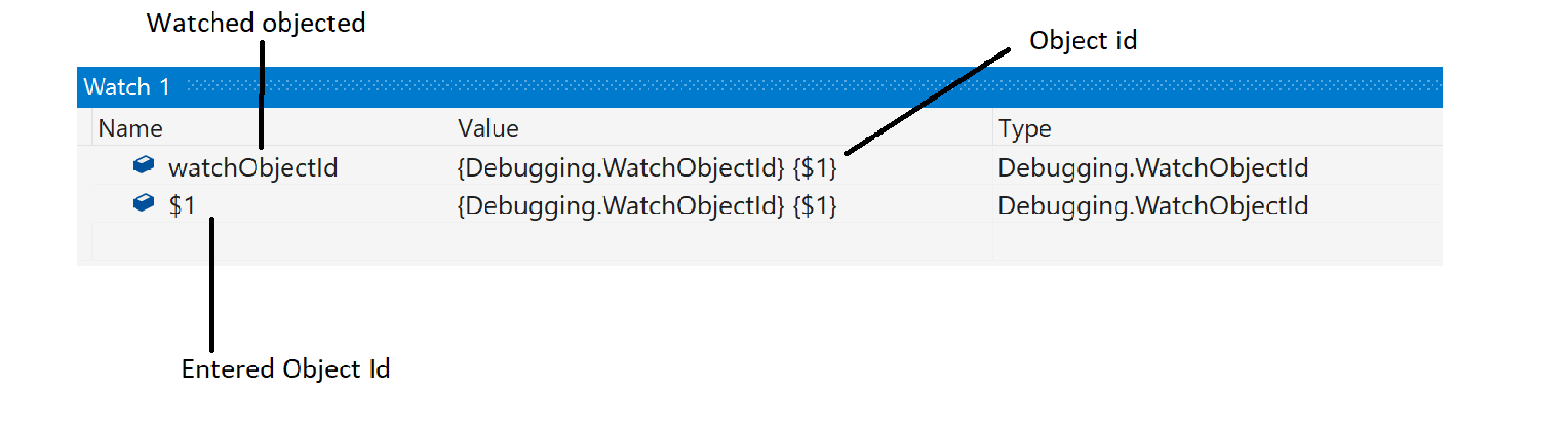
DEBUGGING WINDOWS - CALL STACK



Call Stack

Name	Language	Frame Status
Calculator1.dll!Calculator1.Calculator.Divide(int num1 = 5, int num2 = 0) Line 8	C#	Symbols loaded.
Debugging.exe!Debugging.Program.Conditions() Line 147	C#	Symbols loaded.
Debugging.exe!Debugging.Program.Main(string[] args = {string[0]}) Line 44	C#	Symbols loaded.

DEBUGGING WINDOWS - WATCH WINDOW



Watched objected

Object id

Watch 1

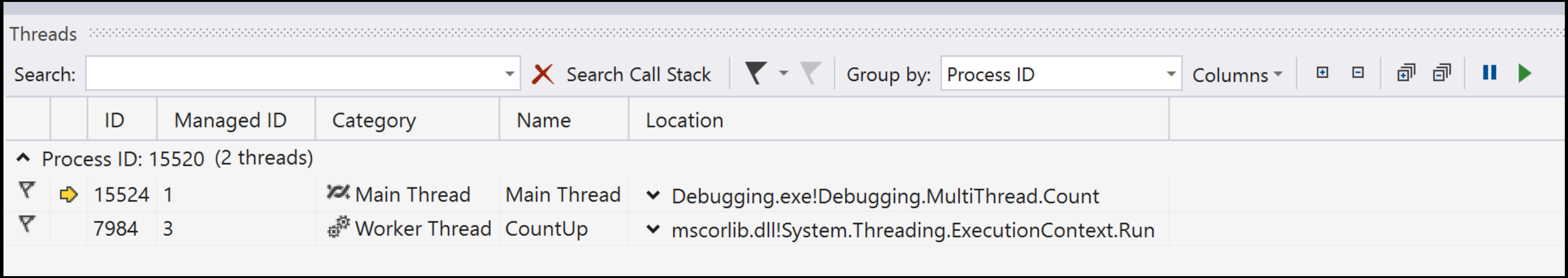
Name	Value	Type
watchObjectId	{Debugging.WatchObjectId} {$1}	Debugging.WatchObjectId
$1	{Debugging.WatchObjectId} {$1}	Debugging.WatchObjectId

Entered Object Id

MULTI-THREADED APPLICATIONS

- Thread Window
- Debug Toolbar
- Debug Location
- Parallel Stacks
- Parallel Watch

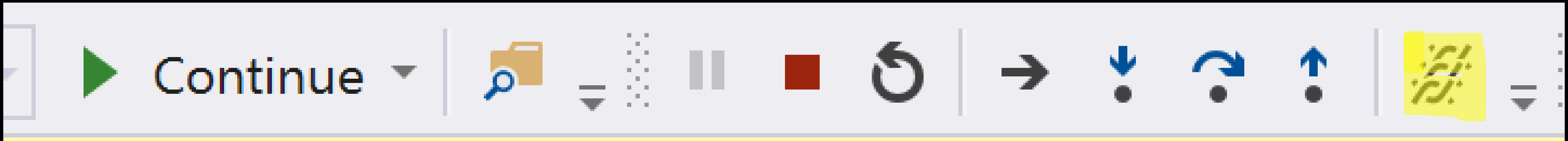
MULTITHREADED DEBUGGING WINDOWS - THREADS

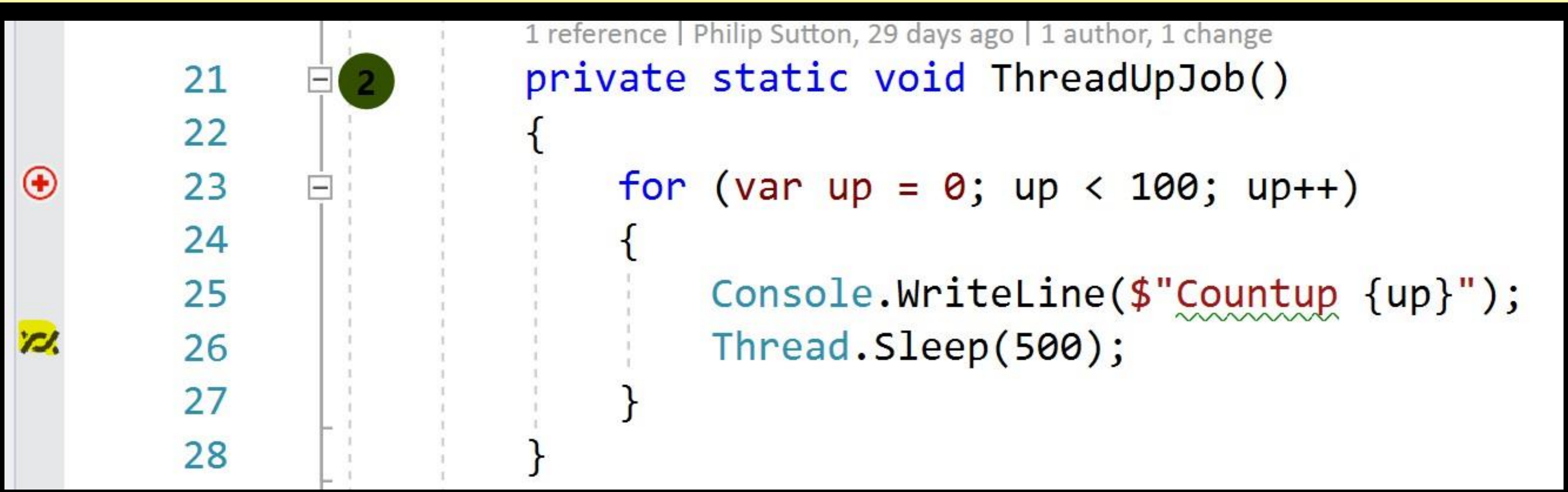


Group by: Process ID

		ID	Managed ID	Category	Name	Location
Process ID: 15520 (2 threads)						
	⇨	15524	1	Main Thread	Main Thread	Debugging.exe!Debugging.MultiThread.Count
		7984	3	Worker Thread	CountUp	mscorlib.dll!System.Threading.ExecutionContext.Run

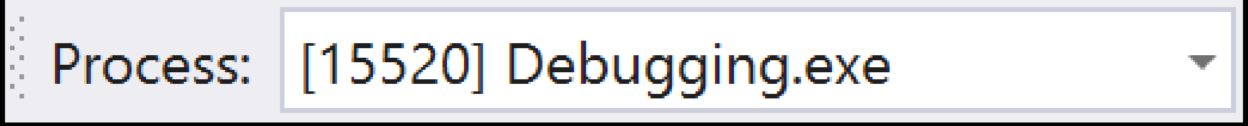
MULTITHREADING WINDOWS – DEBUG TOOLBAR



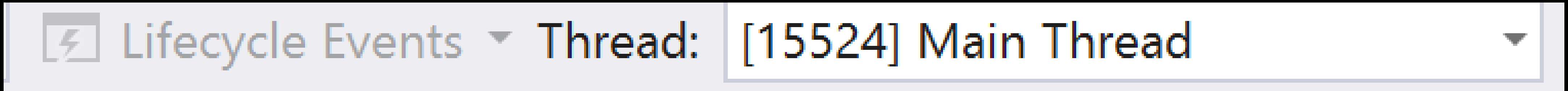

1 reference | Philip Sutton, 29 days ago | 1 author, 1 change

```
private static void ThreadUpJob()
{
    for (var up = 0; up < 100; up++)
    {
        Console.WriteLine($"Countup {up}");
        Thread.Sleep(500);
    }
}
```

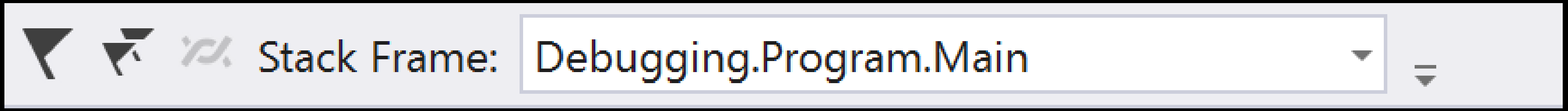
MULTITHREADING WINDOWS
DEBUG LOCATION



Process: [15520] Debugging.exe



Thread: [15524] Main Thread

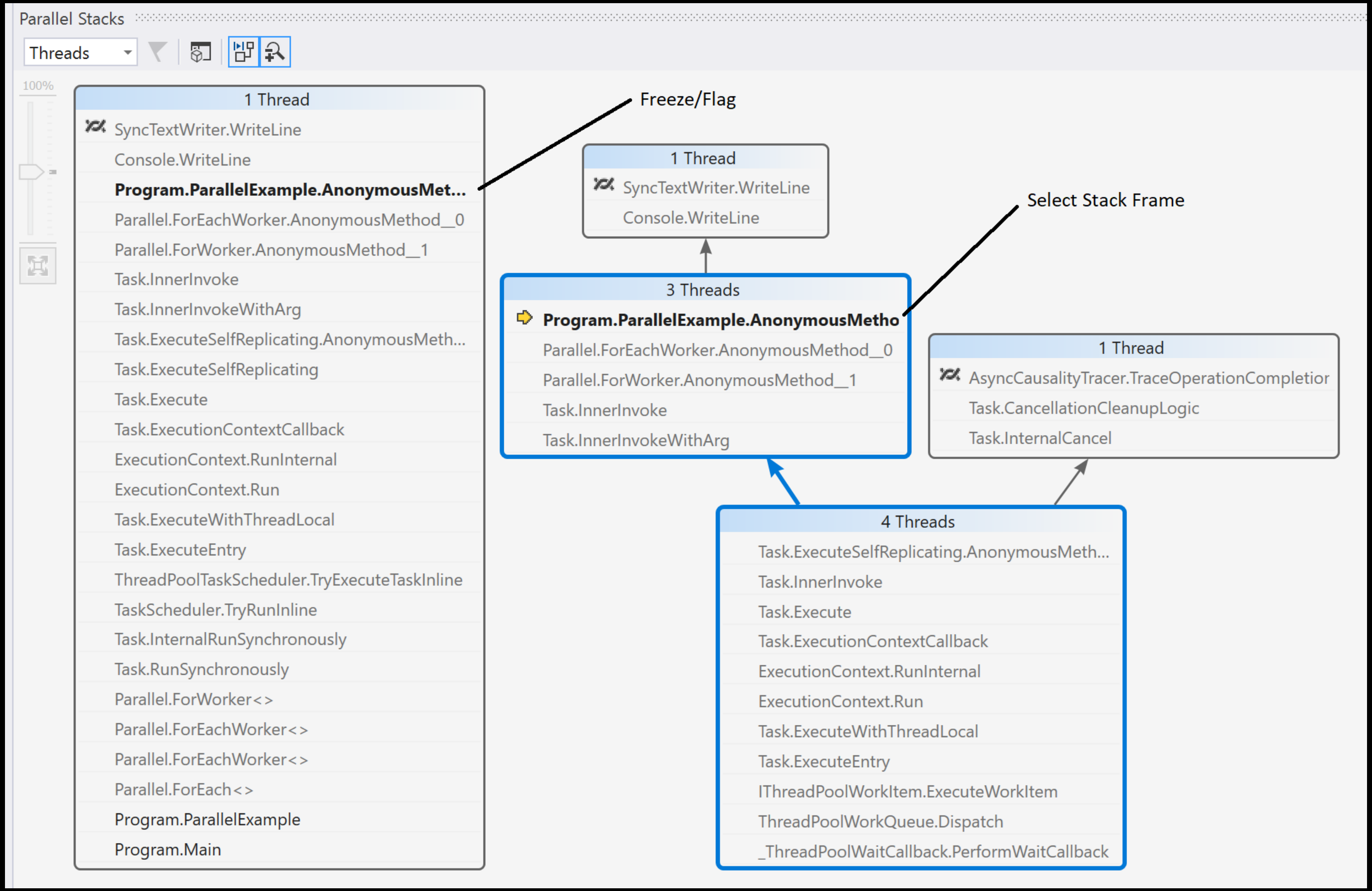


Stack Frame: Debugging.Program.Main

MULTITHREADING WINDOWS PARALLEL STACKS

```
private static void ParallelExample()
    {
        var names = new List<string> {"Bob", "Judy", "Chris", "Ashley"};
        Parallel.ForEach(names, (name) =>
        {
            for (var i = 0; i < 10; i++)
            {
                Console.WriteLine($"Hello {name} {i}");
            }
        });
    }
```

MULTITHREADING WINDOWS PARALLEL STACKS



Freeze/Flag

Select Stack Frame

1 Thread

SyncTextWriter.WriteLine
Console.WriteLine
Program.ParallelExample.AnonymousMet...
Parallel.ForEachWorker.AnonymousMethod__0
Parallel.ForWorker.AnonymousMethod__1
Task.InnerInvoke
Task.InnerInvokeWithArg
Task.ExecuteSelfReplicating.AnonymousMeth...
Task.ExecuteSelfReplicating
Task.Execute
Task.ExecutionContextCallback
ExecutionContext.RunInternal
ExecutionContext.Run
Task.ExecuteWithThreadLocal
Task.ExecuteEntry
ThreadPoolTaskScheduler.TryExecuteTaskInline
TaskScheduler.TryRunInline
Task.InternalRunSynchronously
Task.RunSynchronously
Parallel.ForWorker<>
Parallel.ForEachWorker<>
Parallel.ForEachWorker<>
Parallel.ForEach<>
Program.ParallelExample
Program.Main

1 Thread

SyncTextWriter.WriteLine
Console.WriteLine

3 Threads

Program.ParallelExample.AnonymousMetho
Parallel.ForEachWorker.AnonymousMethod__0
Parallel.ForWorker.AnonymousMethod__1
Task.InnerInvoke
Task.InnerInvokeWithArg

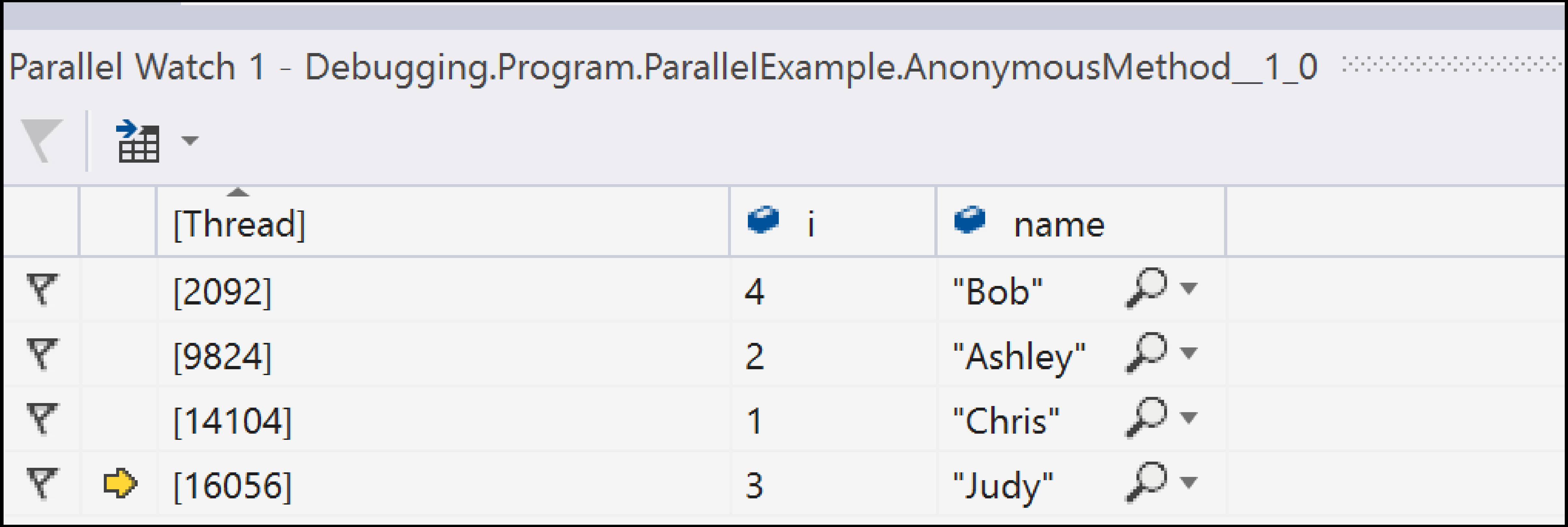
1 Thread

AsyncCausalityTracer.TraceOperationCompletior
Task.CancellationCleanupLogic
Task.InternalCancel

4 Threads

Task.ExecuteSelfReplicating.AnonymousMeth...
Task.InnerInvoke
Task.Execute
Task.ExecutionContextCallback
ExecutionContext.RunInternal
ExecutionContext.Run
Task.ExecuteWithThreadLocal
Task.ExecuteEntry
IThreadPoolWorkItem.ExecuteWorkItem
ThreadPoolWorkQueue.Dispatch
_ThreadPoolWaitCallback.PerformWaitCallback

MULTITHREADING WINDOWS PARALLEL WATCH



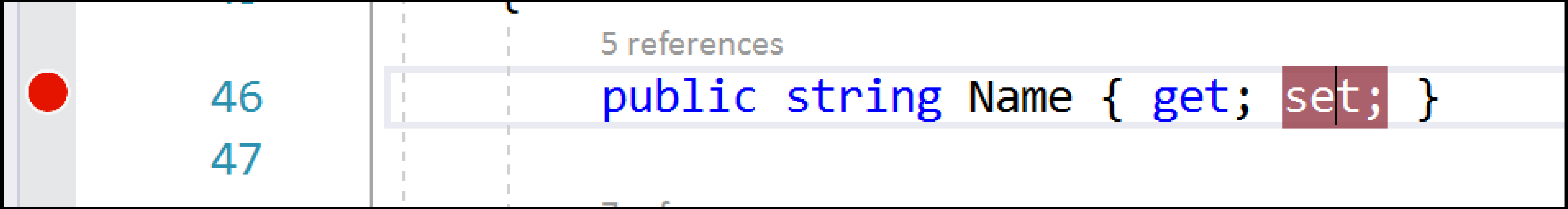
Parallel Watch 1 - Debugging.Program.ParallelExample.AnonymousMethod__1_0

		[Thread]	i	name
		[2092]	4	"Bob"
		[9824]	2	"Ashley"
		[14104]	1	"Chris"
	➡	[16056]	3	"Judy"

BREAKPOINTS

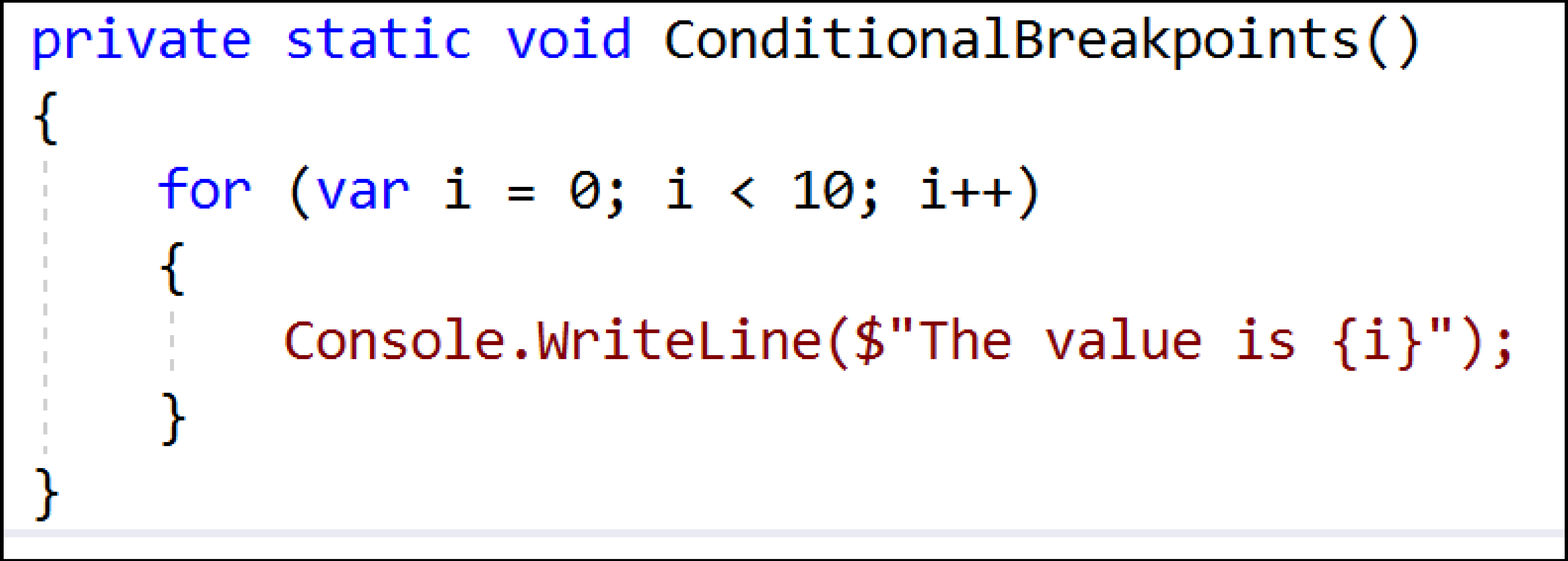
- Data Breakpoints
- Condition
- Hit Count
- Filter
- When hit
- Break at Function

BREAKPOINTS
DATA



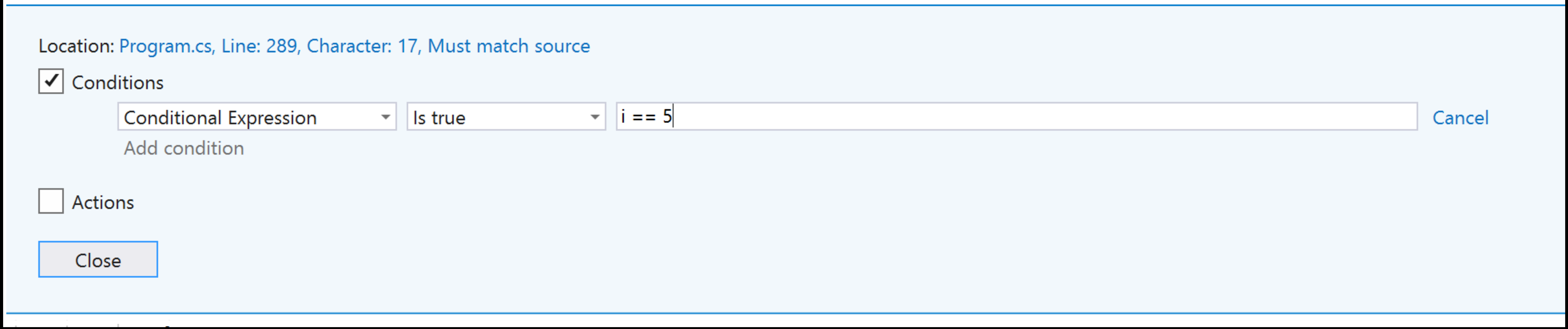
```
5 references
public string Name { get; set; }
```

BREAKPOINTS
CONDITIONAL BREAKPOINTS



```
private static void ConditionalBreakpoints()
{
    for (var i = 0; i < 10; i++)
    {
        Console.WriteLine($"The value is {i}");
    }
}
```

BREAKPOINTS CONDITIONAL



Location: Program.cs, Line: 289, Character: 17, Must match source

☑ Conditions

Conditional Expression | Is true | i == 5

☐ Actions

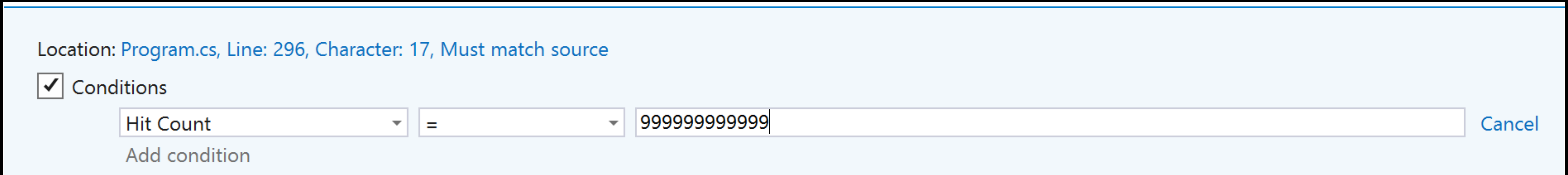
BREAKPOINTS CONDITIONAL

```
private static void ConditionalBreakpoints()
        {
            for (var i = 0; i < 10; i++)
            {
                if (i==5)
                {
                    Debugger.Break();
                }
                Console.WriteLine($"The value is {i}");
            }
        }
```

BREAKPOINTS CONDITIONAL

```
• private static void ConditionalBreakpoints()
•         {
•             for (var i = 0; i < 10; i++)
•             {
•                 if ((Debugger.IsAttached) && (i==5))
•                 {
•                     Debugger.Break();
•                 }
•                 Console.WriteLine($"The value is {i}");
•             }
•         }
```

BREAKPOINTS
HIT COUNT

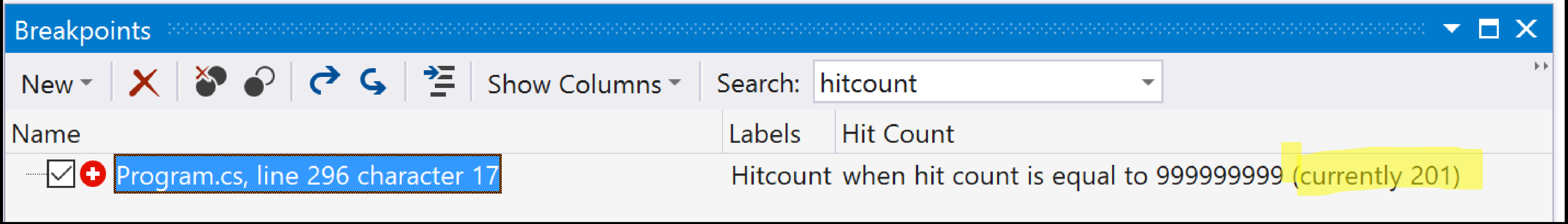


Location: Program.cs, Line: 296, Character: 17, Must match source

☑ Conditions

Hit Count | = | 999999999999

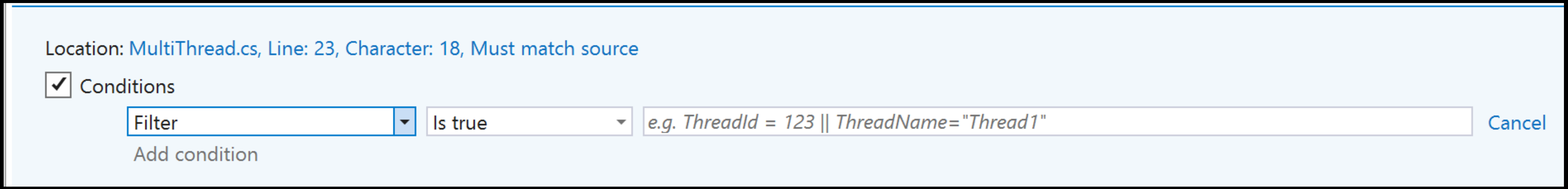
BREAKPOINTS HIT COUNT



Search: hitcount

Name	Labels	Hit Count
☑ Program.cs, line 296 character 17	Hitcount	when hit count is equal to 999999999 (currently 201)

BREAKPOINTS FILTER

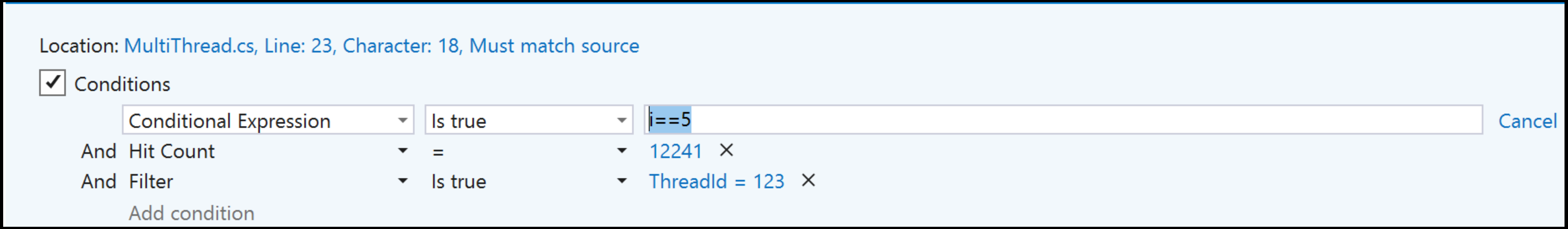


Location: MultiThread.cs, Line: 23, Character: 18, Must match source

☑ Conditions

Filter | Is true | e.g. ThreadId = 123 || ThreadName="Thread1"

BREAKPOINTS FILTER



Location: MultiThread.cs, Line: 23, Character: 18, Must match source

☑ Conditions

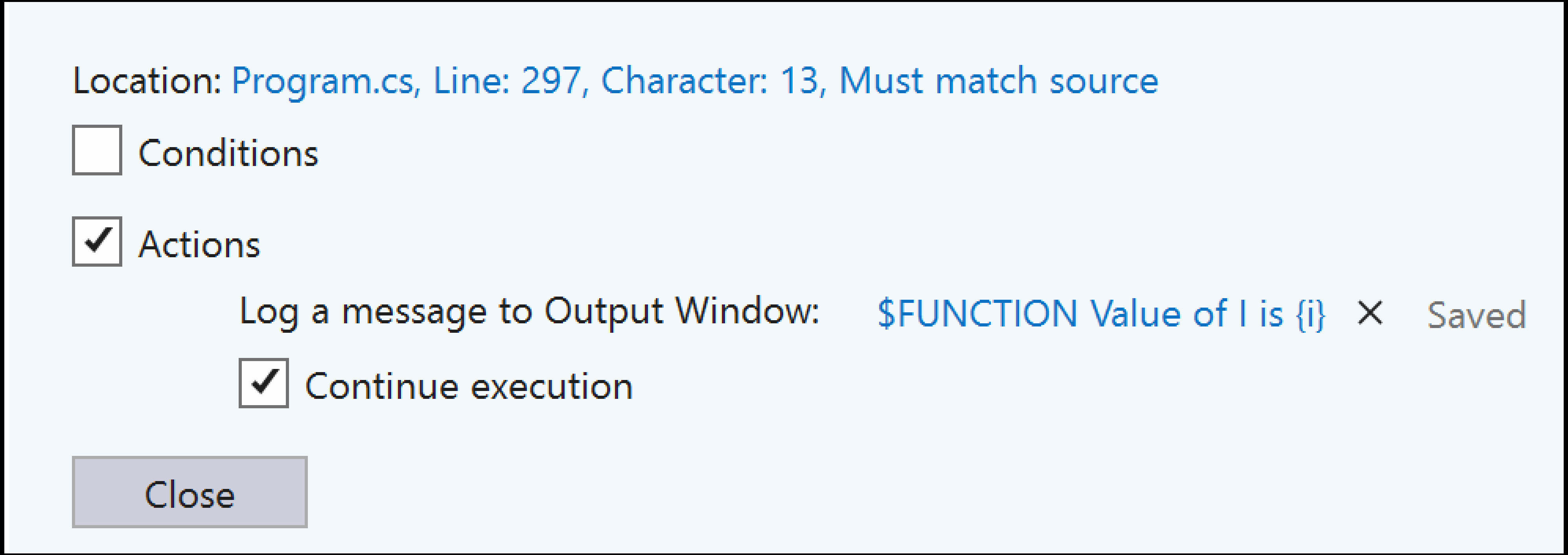
Conditional Expression | Is true | i==5 | Cancel

And Hit Count | = | 12241 ×

And Filter | Is true | ThreadId = 123 ×

Add condition

BREAKPOINTS WHEN HIT



Location: Program.cs, Line: 297, Character: 13, Must match source

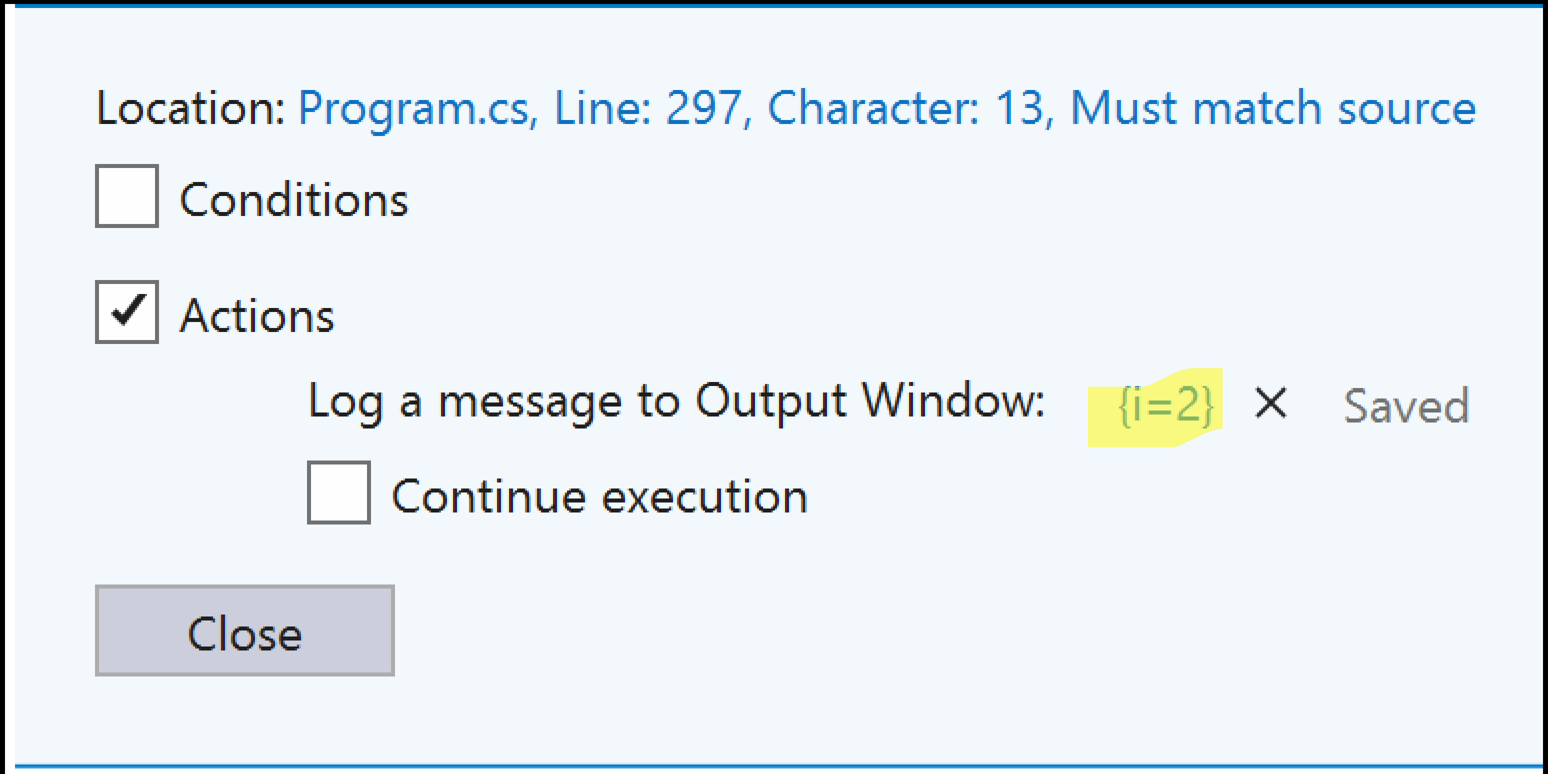
□ Conditions

☑ Actions

Log a message to Output Window: $FUNCTION Value of I is {i} × Saved

☑ Continue execution

BREAKPOINTS WHEN HIT



Location: Program.cs, Line: 297, Character: 13, Must match source

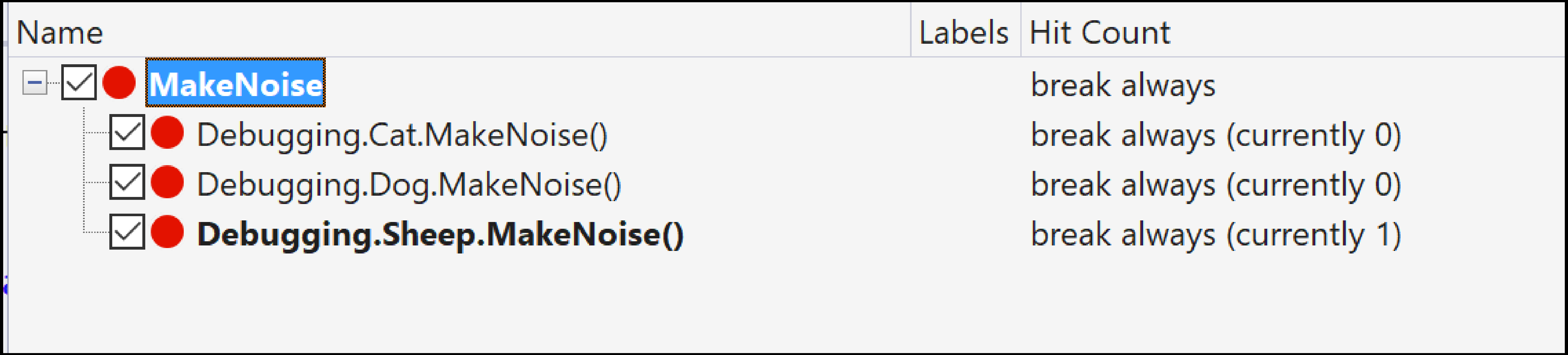
☐ Conditions

☑ Actions

Log a message to Output Window: {i=2}

☐ Continue execution

BREAKPOINTS
BREAK AT FUNCTION



Name	Labels	Hit Count
− ☑ ● MakeNoise		break always
☑ ● Debugging.Cat.MakeNoise()		break always (currently 0)
☑ ● Debugging.Dog.MakeNoise()		break always (currently 0)
☑ ● **Debugging.Sheep.MakeNoise()**		break always (currently 1)

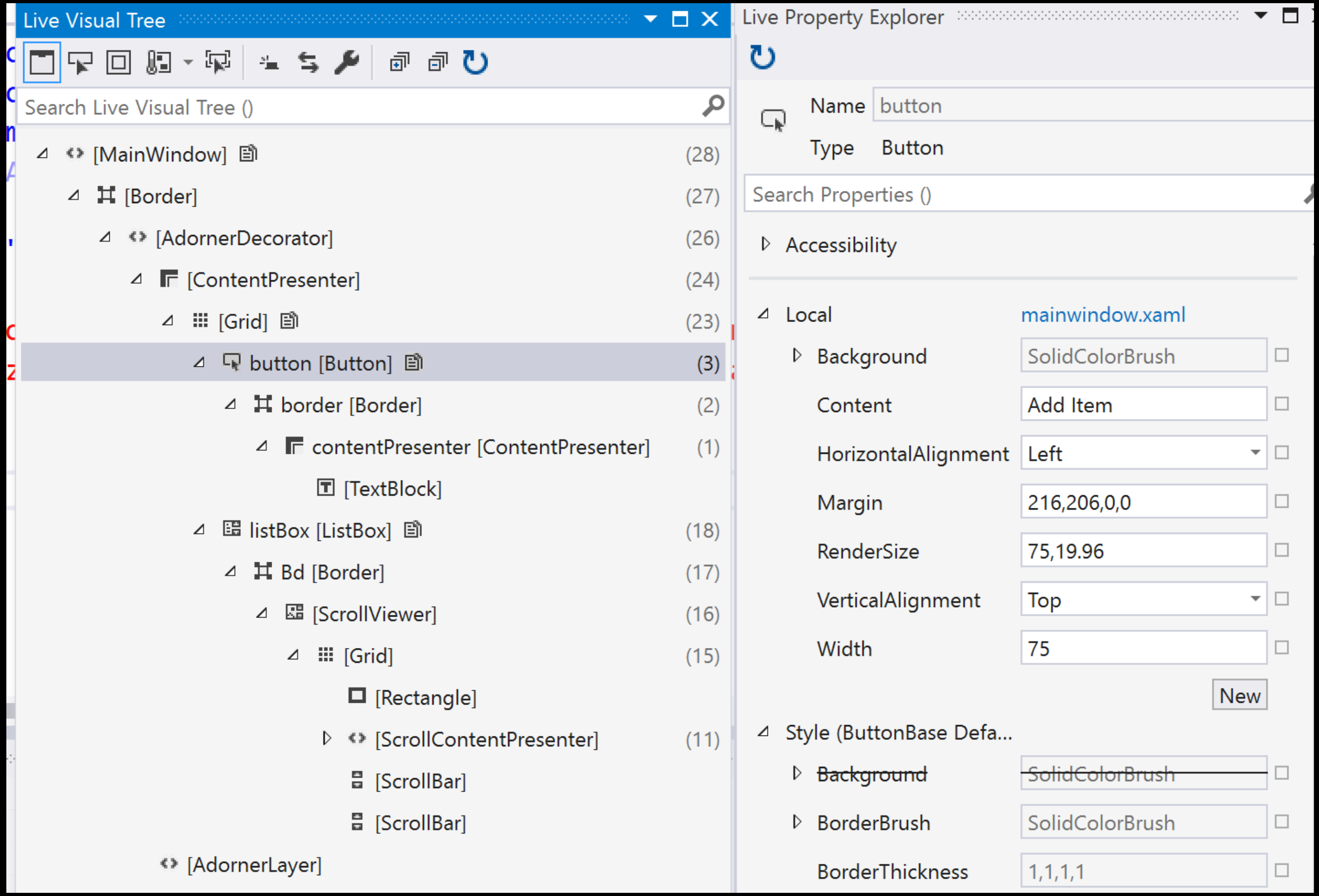
ADDRESS LEVEL DEBUGGING

- Memory
- Disassembly
- Registers

WPF/UWP

- Live Visual Tree
- Live Property Explorer

WPF/UWP
VISUAL TREE/ PROPERTY EXPLORER



```
◿ [MainWindow]                                    (28)
   ◿ [Border]                                     (27)
      ◿ [AdornerDecorator]                        (26)
         ◿ [ContentPresenter]                     (24)
            ◿ [Grid]                              (23)
               ◿ button [Button]                  (3)
                  ◿ border [Border]               (2)
                     ◿ contentPresenter [ContentPresenter] (1)
                          [TextBlock]
               ◿ listBox [ListBox]                (18)
                  ◿ Bd [Border]                   (17)
                     ◿ [ScrollViewer]             (16)
                        ◿ [Grid]                  (15)
                             [Rectangle]
                           ▷ [ScrollContentPresenter] (11)
                             [ScrollBar]
                             [ScrollBar]
            [AdornerLayer]
```

Name button

Type Button

▷ Accessibility

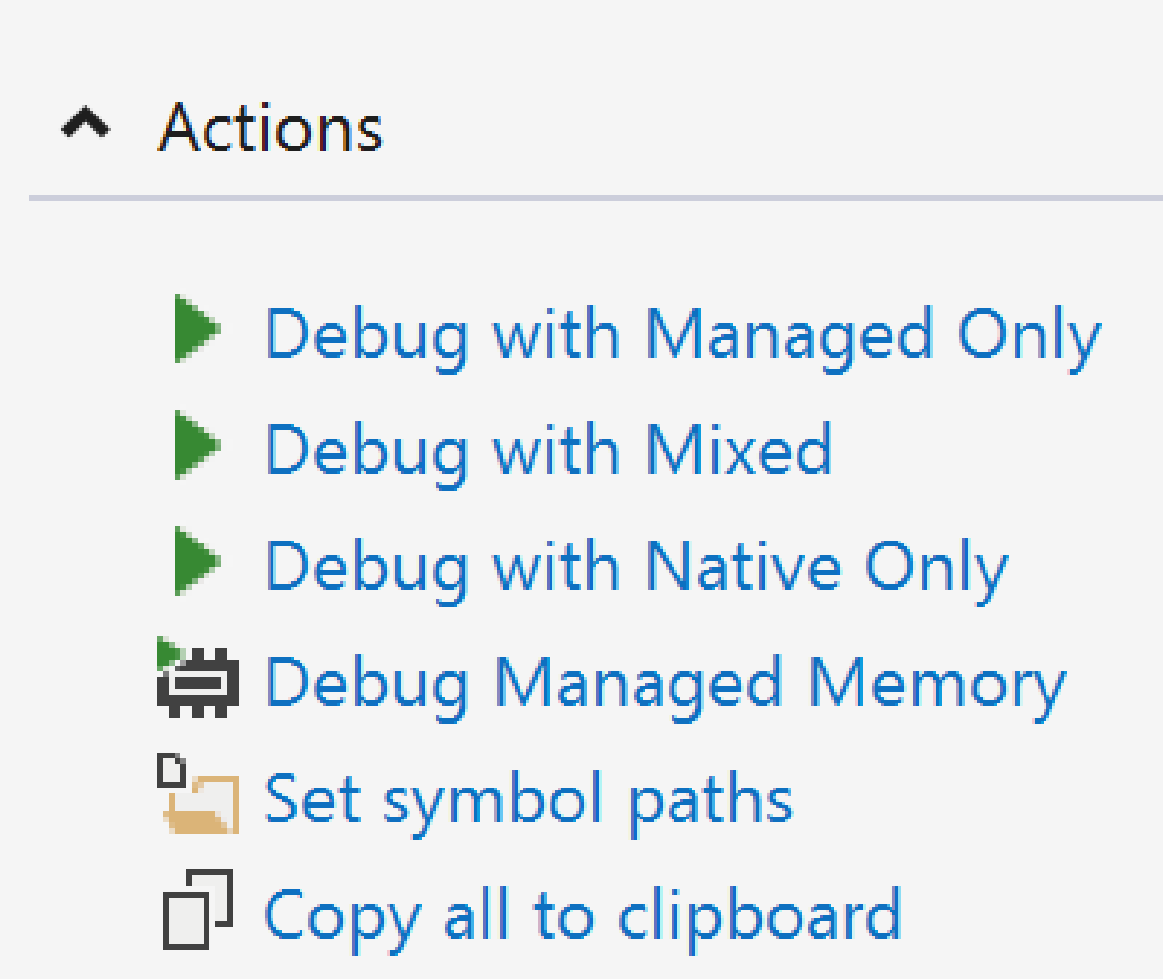
◿ Local mainwindow.xaml

▷ Background	SolidColorBrush	□
Content	Add Item	□
HorizontalAlignment	Left	□
Margin	216,206,0,0	□
RenderSize	75,19.96	□
VerticalAlignment	Top	□
Width	75	□

◿ Style (ButtonBase Defa...

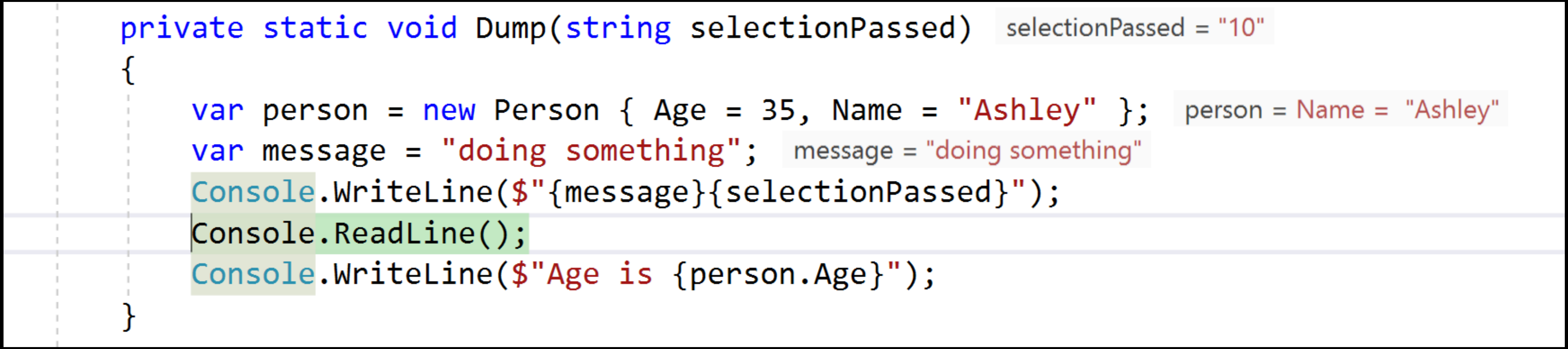
▷ ~~Background~~	~~SolidColorBrush~~	□
▷ BorderBrush	SolidColorBrush	□
BorderThickness	1,1,1,1	□

DUMPS



^ Actions
Debug with Managed Only
Debug with Mixed
Debug with Native Only
Debug Managed Memory
Set symbol paths
Copy all to clipboard

DUMPS



```
private static void Dump(string selectionPassed)  selectionPassed = "10"
{
    var person = new Person { Age = 35, Name = "Ashley" };  person = Name =  "Ashley"
    var message = "doing something";  message = "doing something"
    Console.WriteLine($"{message}{selectionPassed}");
    Console.ReadLine();
    Console.WriteLine($"Age is {person.Age}");
}
```

SYMBOL SERVERS

- The symbol server enables the debuggers to automatically retrieve the correct symbol files without product names, releases, or build numbers.

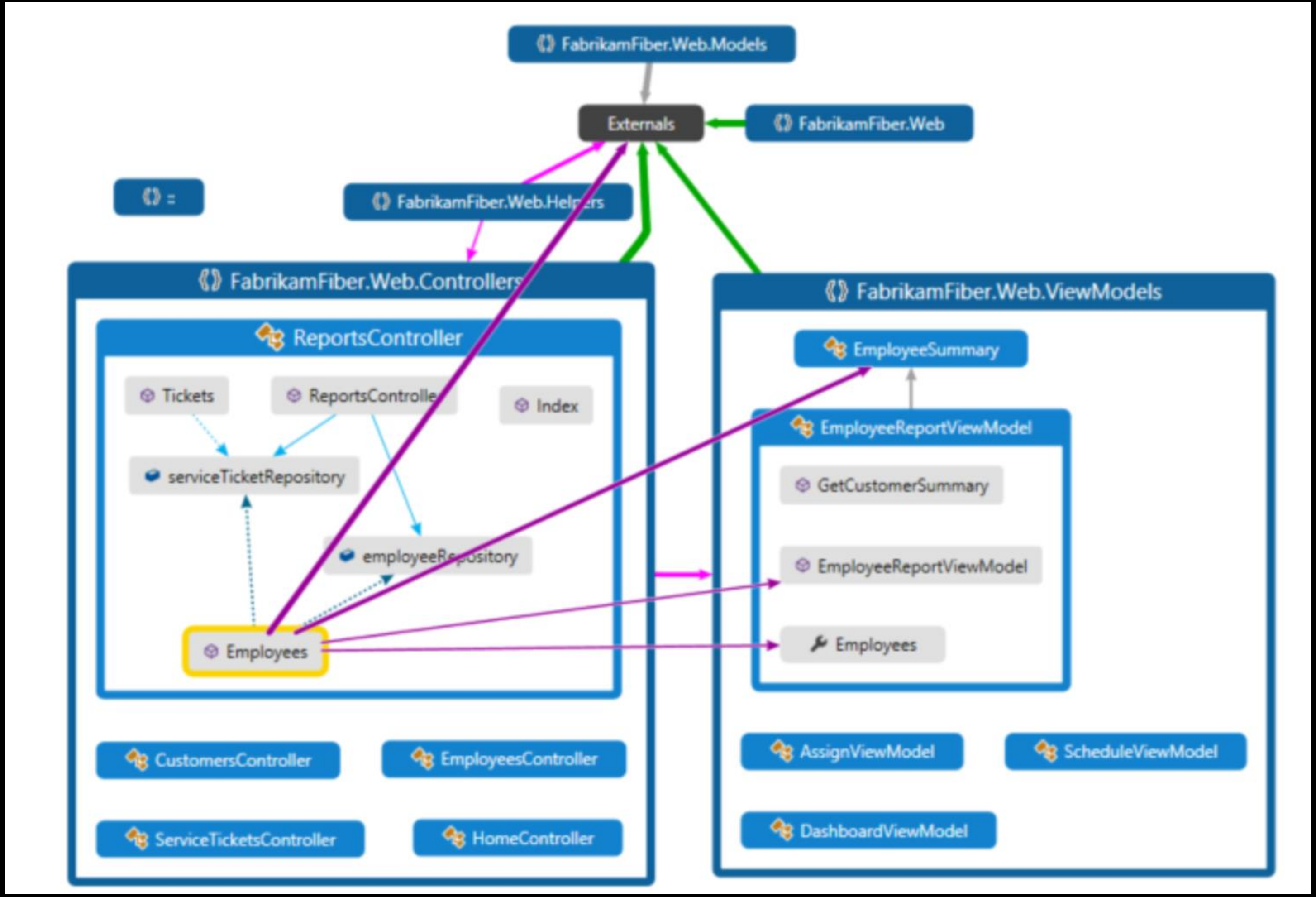
FRONT END

- Javascript
- Browser Link
- Sync
- Inspector

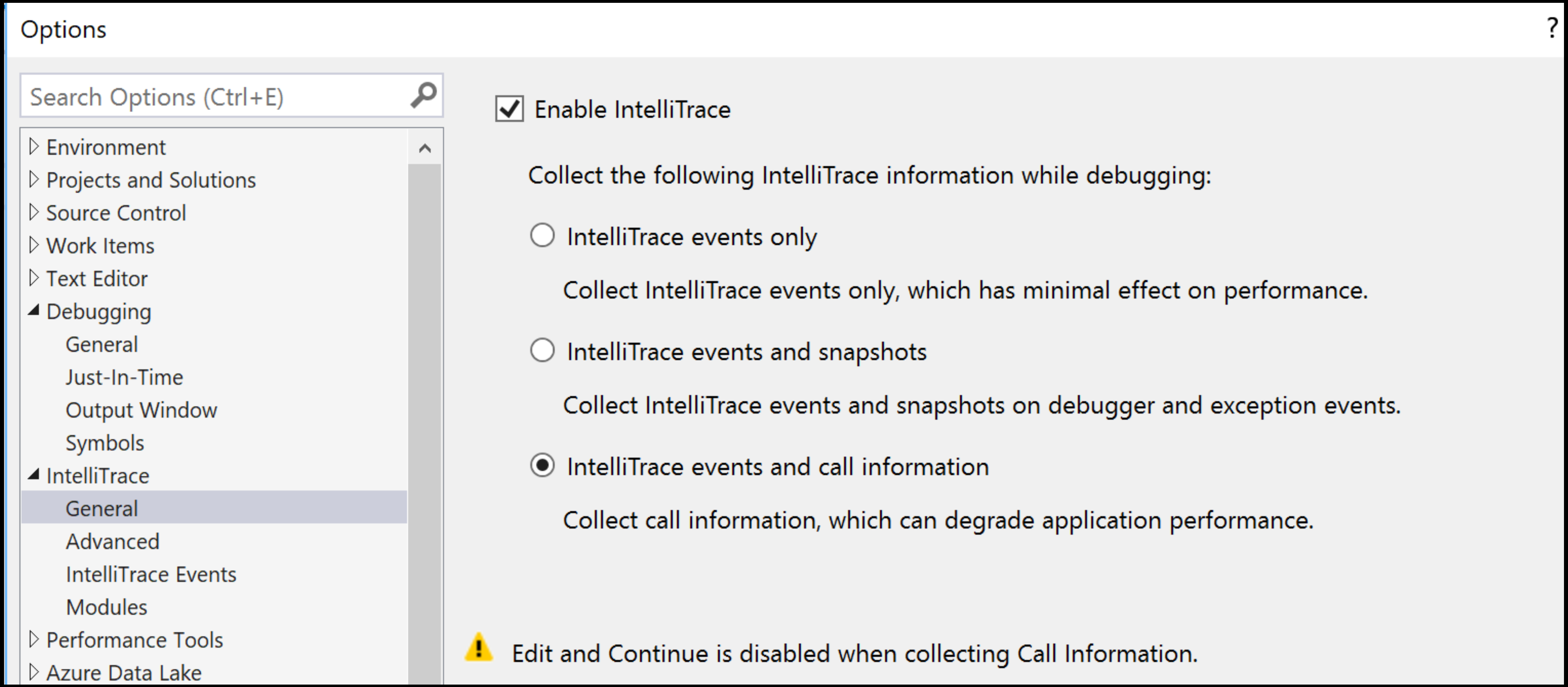
ENTERPRISE

- Code Maps
- Intellitrace

ENTERPRISE CODE MAPS



ENTERPRISE INTELLITRACE



☑ Enable IntelliTrace

Collect the following IntelliTrace information while debugging:

○ IntelliTrace events only

Collect IntelliTrace events only, which has minimal effect on performance.

○ IntelliTrace events and snapshots

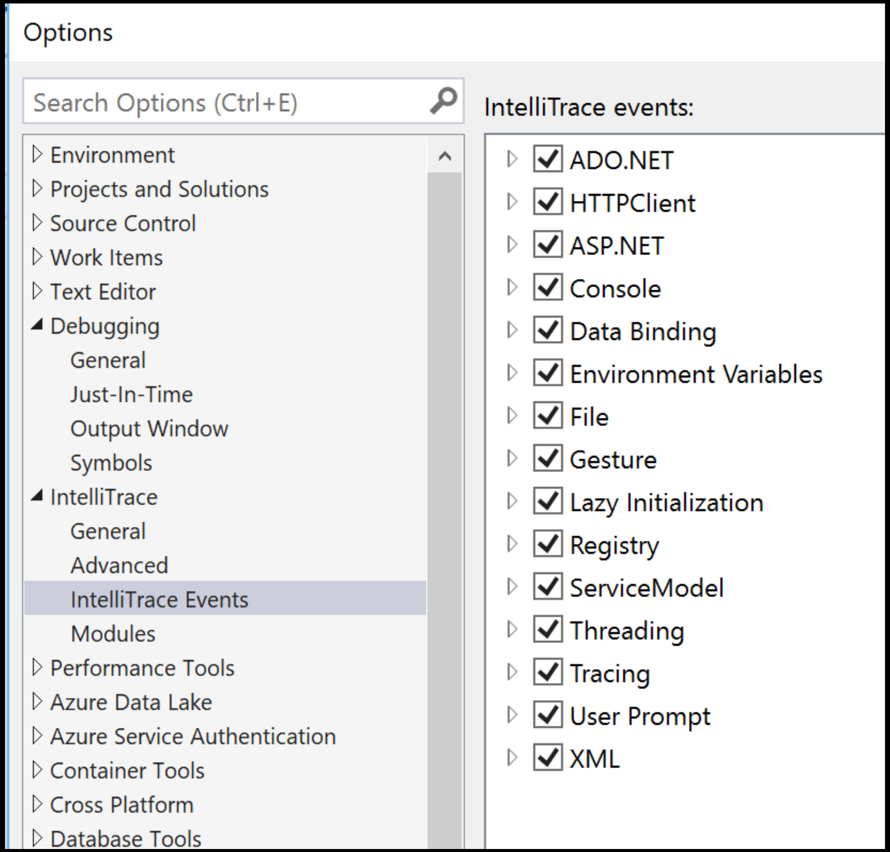
Collect IntelliTrace events and snapshots on debugger and exception events.

● IntelliTrace events and call information

Collect call information, which can degrade application performance.

⚠ Edit and Continue is disabled when collecting Call Information.

ENTERPRISE INTELLITRACE



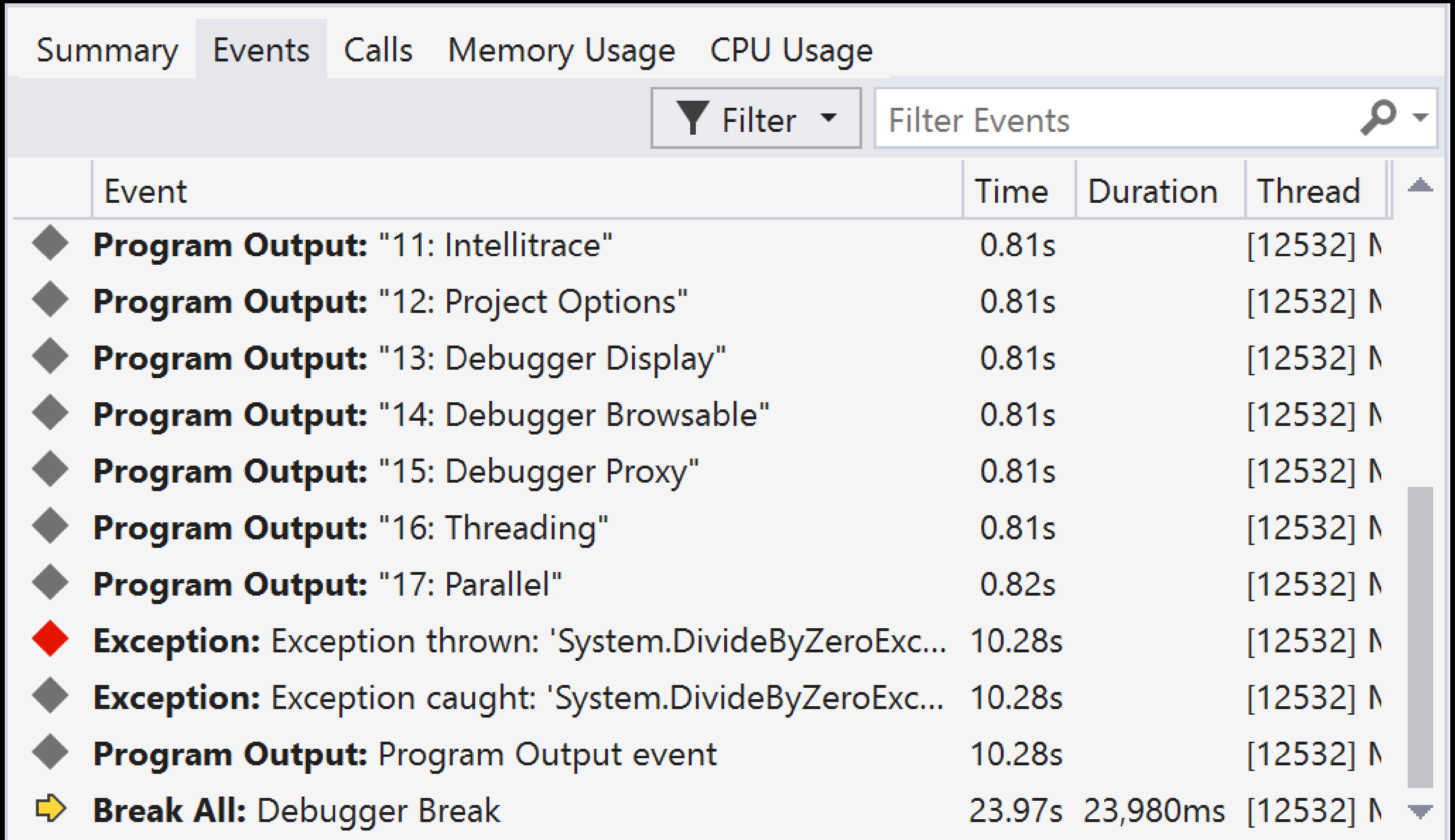
Options

Search Options (Ctrl+E)

IntelliTrace events:

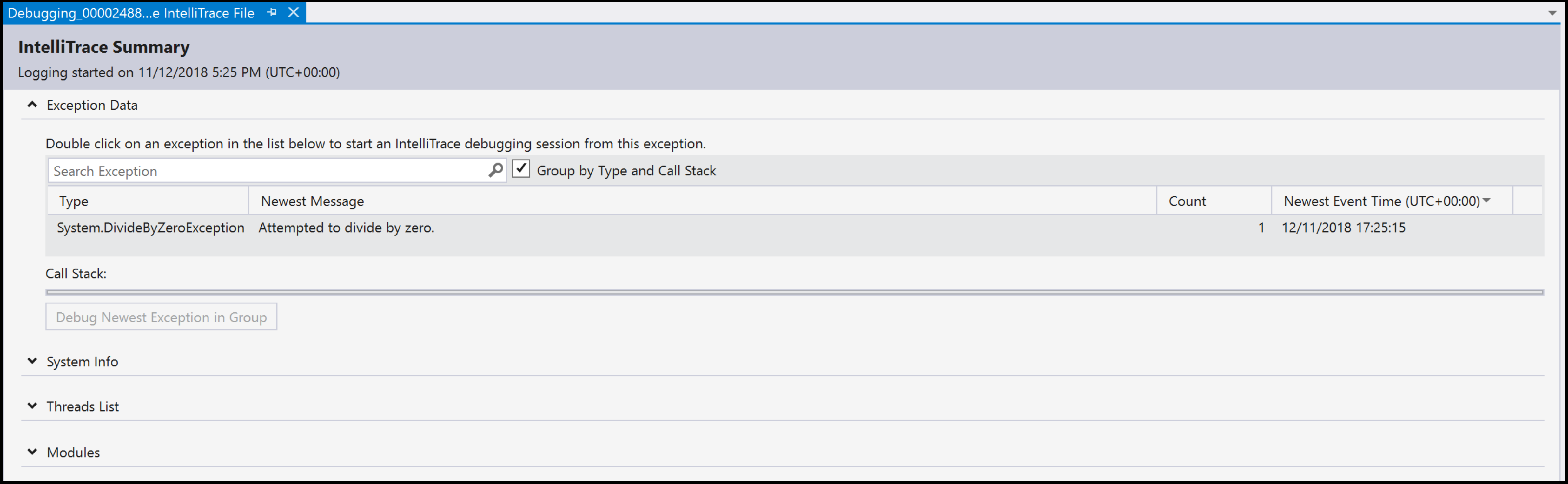
- ☑ ADO.NET
- ☑ HTTPClient
- ☑ ASP.NET
- ☑ Console
- ☑ Data Binding
- ☑ Environment Variables
- ☑ File
- ☑ Gesture
- ☑ Lazy Initialization
- ☑ Registry
- ☑ ServiceModel
- ☑ Threading
- ☑ Tracing
- ☑ User Prompt
- ☑ XML

ENTERPRISE INTELLITRACE



	Event	Time	Duration	Thread
◆	**Program Output:** "11: Intellitrace"	0.81s		[12532] N
◆	**Program Output:** "12: Project Options"	0.81s		[12532] N
◆	**Program Output:** "13: Debugger Display"	0.81s		[12532] N
◆	**Program Output:** "14: Debugger Browsable"	0.81s		[12532] N
◆	**Program Output:** "15: Debugger Proxy"	0.81s		[12532] N
◆	**Program Output:** "16: Threading"	0.81s		[12532] N
◆	**Program Output:** "17: Parallel"	0.82s		[12532] N
◆	**Exception:** Exception thrown: 'System.DivideByZeroExc...	10.28s		[12532] N
◆	**Exception:** Exception caught: 'System.DivideByZeroExc...	10.28s		[12532] N
◆	**Program Output:** Program Output event	10.28s		[12532] N
⇨	**Break All:** Debugger Break	23.97s	23,980ms	[12532] N

ENTERPRISE INTELLITRACE



IntelliTrace Summary

Logging started on 11/12/2018 5:25 PM (UTC+00:00)

Exception Data

Double click on an exception in the list below to start an IntelliTrace debugging session from this exception.

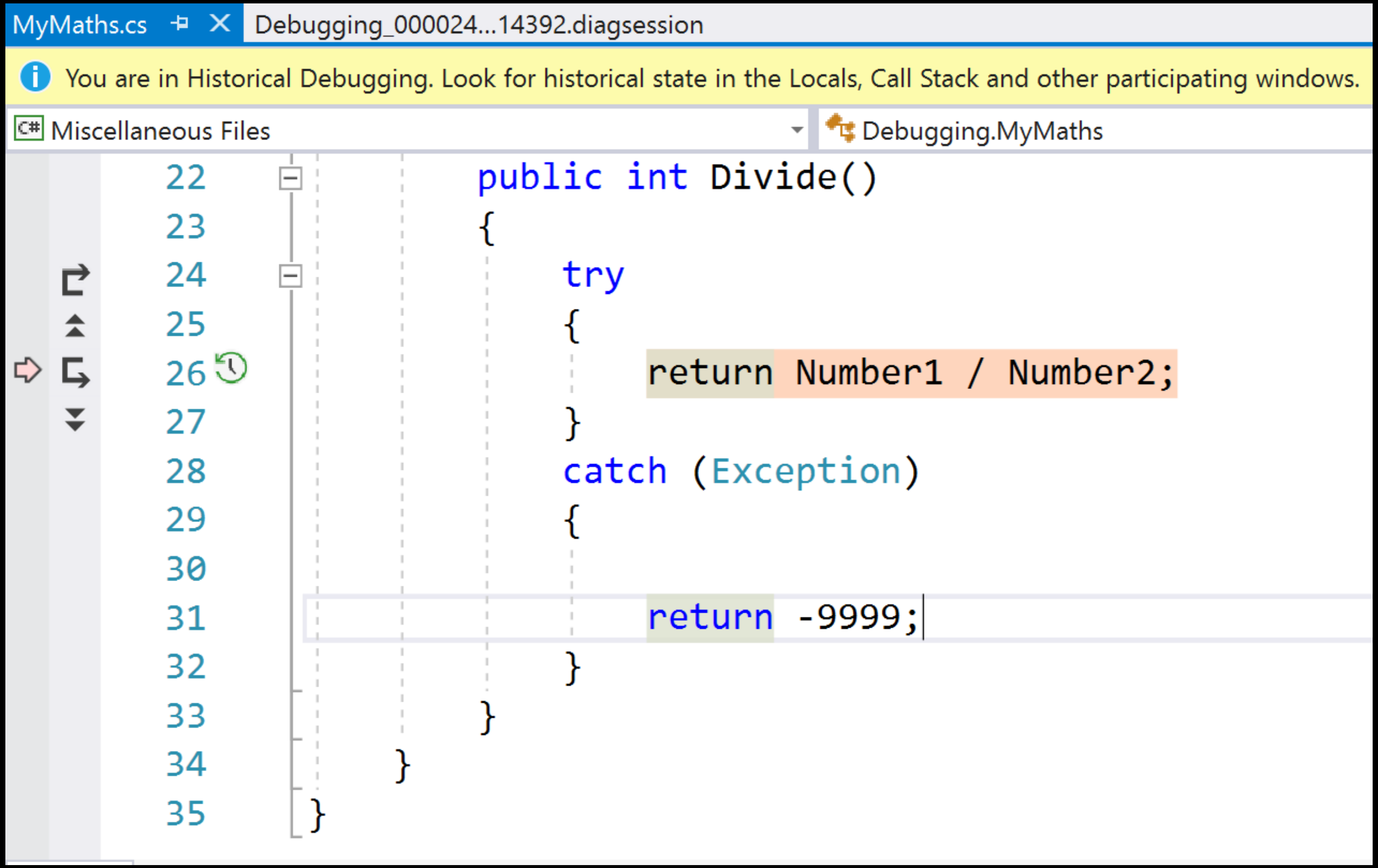
Search Exception

☑ Group by Type and Call Stack

Type	Newest Message	Count	Newest Event Time (UTC+00:00)
System.DivideByZeroException	Attempted to divide by zero.	1	12/11/2018 17:25:15

Call Stack:

ENTERPRISE INTELLITRACE



You are in Historical Debugging. Look for historical state in the Locals, Call Stack and other participating windows.

```
        public int Divide()
        {
            try
            {
                return Number1 / Number2;
            }
            catch (Exception)
            {

                return -9999;
            }
        }
    }
}
```

OTHER DEBUGGING ITEMS

- Remote Debugging
- Azure Debugging
- Live Sharing
- WinDbg
- SOS

REFERENCES

- https://docs.microsoft.com/en-us/visualstudio/debugger - general debugging
- https://docs.microsoft.com/en-us/visualstudio/debugger/general-debugging-options-dialog-box?view=vs-2017 debugging options
- https://www.wintellect.com/pdb-files-what-every-developer-must-know/ pdb files
- https://blogs.msdn.microsoft.com/jaredpar/2011/03/18/debuggerdisplay-attribute-best-practices/ (old) debugger display best practices
- https://blogs.msdn.microsoft.com/visualstudio/2017/06/26/7-lesser-known-hacks-for-debugging-in-visual-studio/

QUESTIONS