**.NET Debugging Demos Lab 1: Hang**

This is the first in a series of about 10 labs on .NET debugging. The lab will use a site called BuggyBits, and as the name suggests the bits are extremely buggy.

**Note:** The questions in the labs are only meant as an aid when troubleshooting the problem. I will moderate any comments containing answers to these questions until I have released the lab review (about a week after the original lab post in order to give everyone a shot at the labs without answers)

Feel free to comment on the lab format good or bad so that I know what works well and what doesn’t for future labs.

Without further ado, here comes Lab 1

**Reproduce the problem**

1. Browse to Featured Products (https://localhost:44350/Products/Featured).

This should take about 5 seconds to show, you can see the start time and execution time the bottom of the page.

1. Open up 5 browsers, all browsing to this site and refresh them simultaneously

Note the execution time for each of them and make sure that the start time is pretty much the same on all (otherwise you probably didn’t run the reg file)

Questions:

* What are the execution times?
* What is the CPU usage of the w3wp.exe process when reproducing the problem? High or low CPU?
* What are some potential reasons for a hang with these symptoms?

**Get a memory dump**

1. Start a command window and browse to your debuggers directory. Type the following command to prepare taking the dump but don’t hit enter quite yet.
2. procdump64.exe -ma iisexpress.exe

or with dotnet-dump

dotnet-dump -n iisexpress.exe

1. Reproduce the problem either by browsing with 5 browsers as you did before or by stressing the site with tinyget with the following command line
2. .\ToolsAndScripts\tinyget.ps1 -url https://localhost:44350/Products/Featured -numTimes 100
3. Hit enter in the procdump/dotnet-dump command window to take the memory dump while the requests are still executing.

Questions:

* What triggers the generation of the memory dump?
* What permissions do you need to take a memory dump of a process?
* What is the difference between running with -ma or without?

**Open the dump in Windbg**

1. Open windbg and open the memory dump (.dmp file) with File/Open Crash dump.
2. Set up the symbol path (see Information and Setup Instructions for more info)
3. Load sos (see Information and Setup Instructions for more info)

**Examine the stacks**

1. Examine the native call stacks
2. ~\* kb 2000
3. Examine the .net call stacks
4. ~\* e !clrstack

Questions:

* Do you see any patterns or recognize any of the call stacks that suggests a thread is waiting for a synchronization mechanism?

**Troubleshoot the hang**

1. Determine the ID of the thread owning the lock
2. !syncblk
   * What thread owns the lock?
   * How many threads are waiting for the lock?

Hint: MonitorHeld = 1 for each owner and 2 for each waiter.

1. Pick one of the waiters (Hint: waiters will sit in AwareLock::Enter) and take a look at what it is doing.
2. ~5s (move to thread 5, replace 5 with actual thread ID)
3. kb 2000 (examine native stack)
4. !clrstack (examine .net stack)
   * In which .net function is it waiting for the lock?
5. Determine what the owning thread is doing
6. ~5s (move to thread 5, replace 5 with actual thread ID)
7. kb 2000 (examine native stack)
8. !clrstack (examine .net stack)
   * Why is it blocking?
9. Examine the code for .NET method owning the lock to verify your theory.