DOTNEXT

Automate your Dumps 💩

Dina Goldshtein
Riverbed

dinazil@gmail.com
@dinagozil

riverbed®

Agenda

- Some motivation
- What’s a dump?
- Getting a dump
- Manually analyzing dumps
- Automating dump analysis

Agenda

- Some motivation
- What's a dump?
- Getting a dump
- Manually analyzing dumps
- Automating dump analysis

} × 2 (Windows, Linux)

Production is Special

- Some things only happen in production environment

Production is Special

- Some things only happen in production environment
- Client’s environment is different
  - OS, service packs, locale
  - Network, configurations, other applications

Production is Special

- Some things only happen in production environment
- Client’s environment is different
  - OS, service packs, locale
  - Network, configurations, other applications
- Simulated input is different from real input

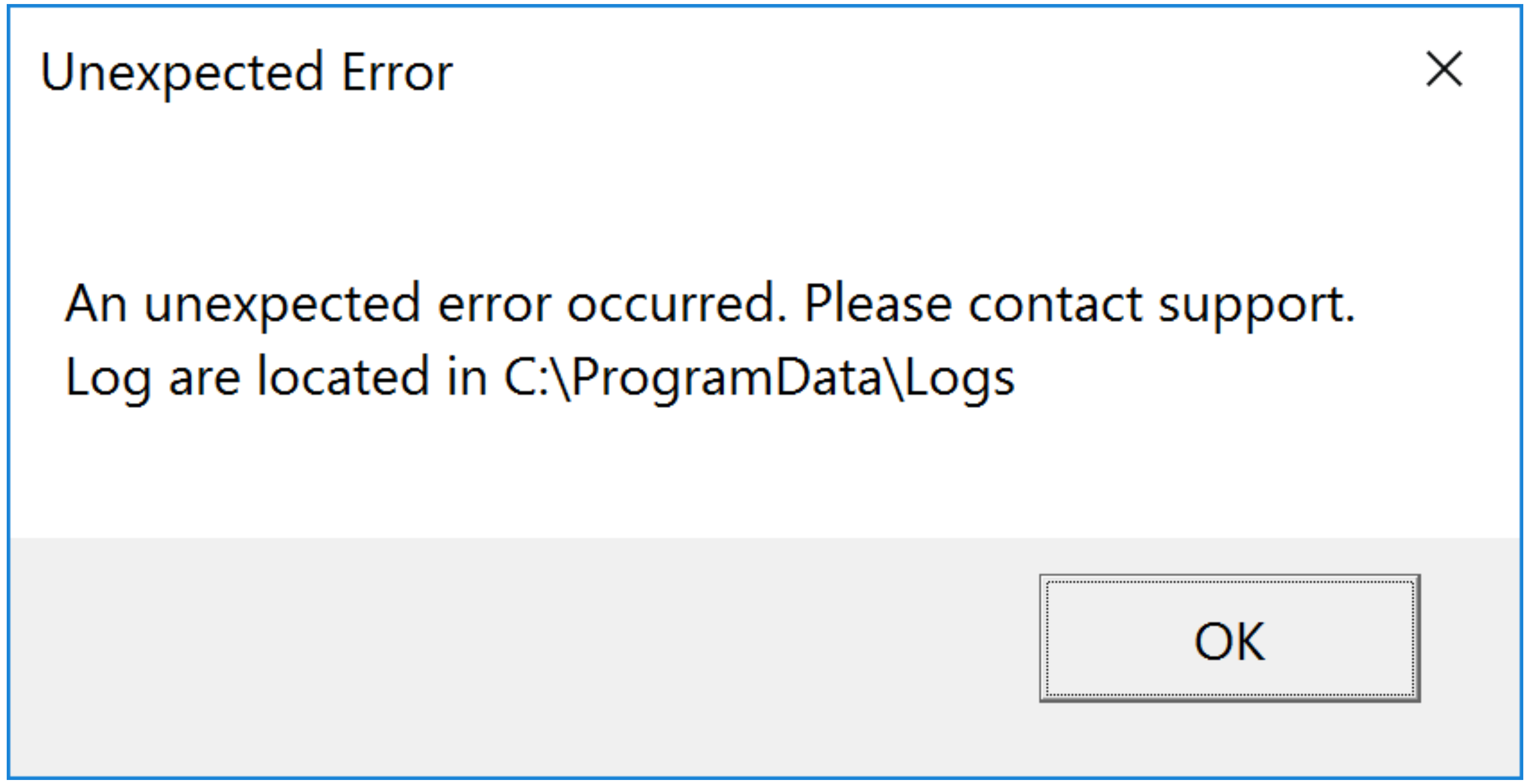
Production is Special

- Some things only happen in production environment
- Client’s environment is different
  - OS, service packs, locale
  - Network, configurations, other applications
- Simulated input is different from real input
- Different system uptime, some bugs are just very rare

In the Beginning…

In the Beginning…

- Mask all crashes by a nice error dialog and an “orderly” shut-down
- Analyze errors using very extensive log files from all components


Unexpected Error

An unexpected error occurred. Please contact support.
Log are located in C:\ProgramData\Logs

In the Beginning…

- Mask all crashes by a nice error dialog and an “orderly” shut-down
- Analyze errors using very extensive log files from all components
- Native exceptions are not caught by managed
- Last error in log doesn’t always correspond to the faulting module

In the Beginning…

- Mask all crashes by a nice error dialog and an “orderly” shut-down
- Analyze errors using very extensive log files from all components
- Native exceptions are not caught by managed
- Last error in log doesn’t always correspond to the fiend
- Exception doesn’t always contain all the needed data (`KeyNotFoundException`)

In the Beginning…

- Mask all crashes by a nice error dialog and an “orderly” shut-down
- Analyze errors using very extensive log files from all components
- Native exceptions are not caught by managed
- Last error in log doesn’t always correspond to the fiend
- Exception doesn’t always contain all the needed data (`KeyNotFoundException`)
- **Need to know exact exception, when it occurred and where!**

Dumps to the Rescue

- A dump is a snapshot of a process's memory
    - Threads
    - Heap
    - Exceptions
    - Locks

Dumps to the Rescue

- A dump is a snapshot of a process's memory
  - Threads
  - Heap
  - Exceptions
  - Locks
- Various tools can open dump files and see what's inside, starting from Visual Studio

What’s in a Dump?

DEMO

Getting A Dump

Capturing Dumps

- Crash dump

Capturing Dumps

- Crash dump
- Just-for-fun dump
    - Memory leak
    - Hang
    - Sudden burst of threads

Capturing Dumps

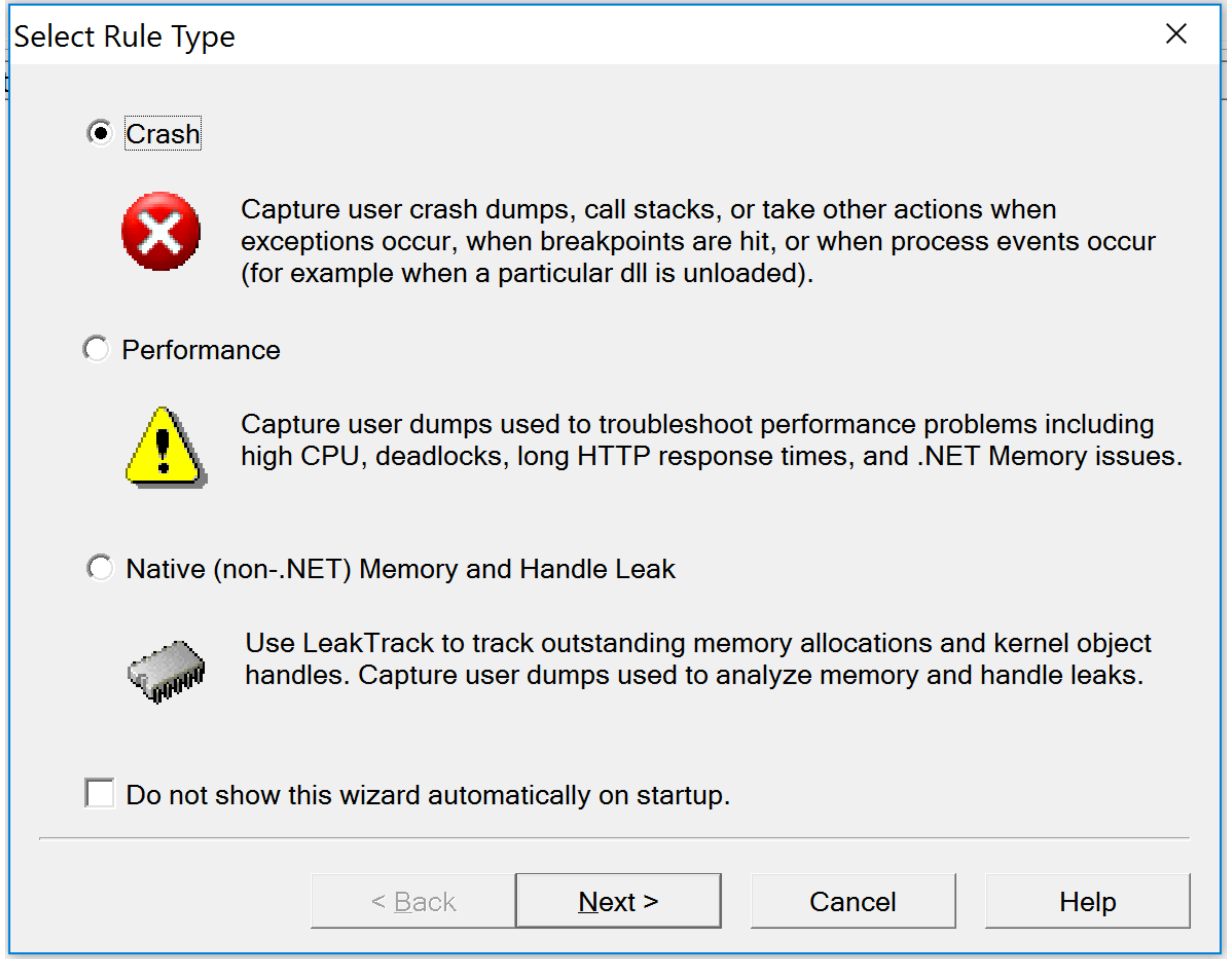
- Crash dump
- Just-for-fun dump
    - Memory leak
    - Hang
    - High CPU

- On Windows you can use:
    - WER Registry key
    - Sysinternals ProcDump
    - DebugDiag
- On Linux you can use:
    - gcore
    - core_pattern

Windows

DEMO

DebugDiag



Select Rule Type

● Crash

Capture user crash dumps, call stacks, or take other actions when exceptions occur, when breakpoints are hit, or when process events occur (for example when a particular dll is unloaded).

○ Performance

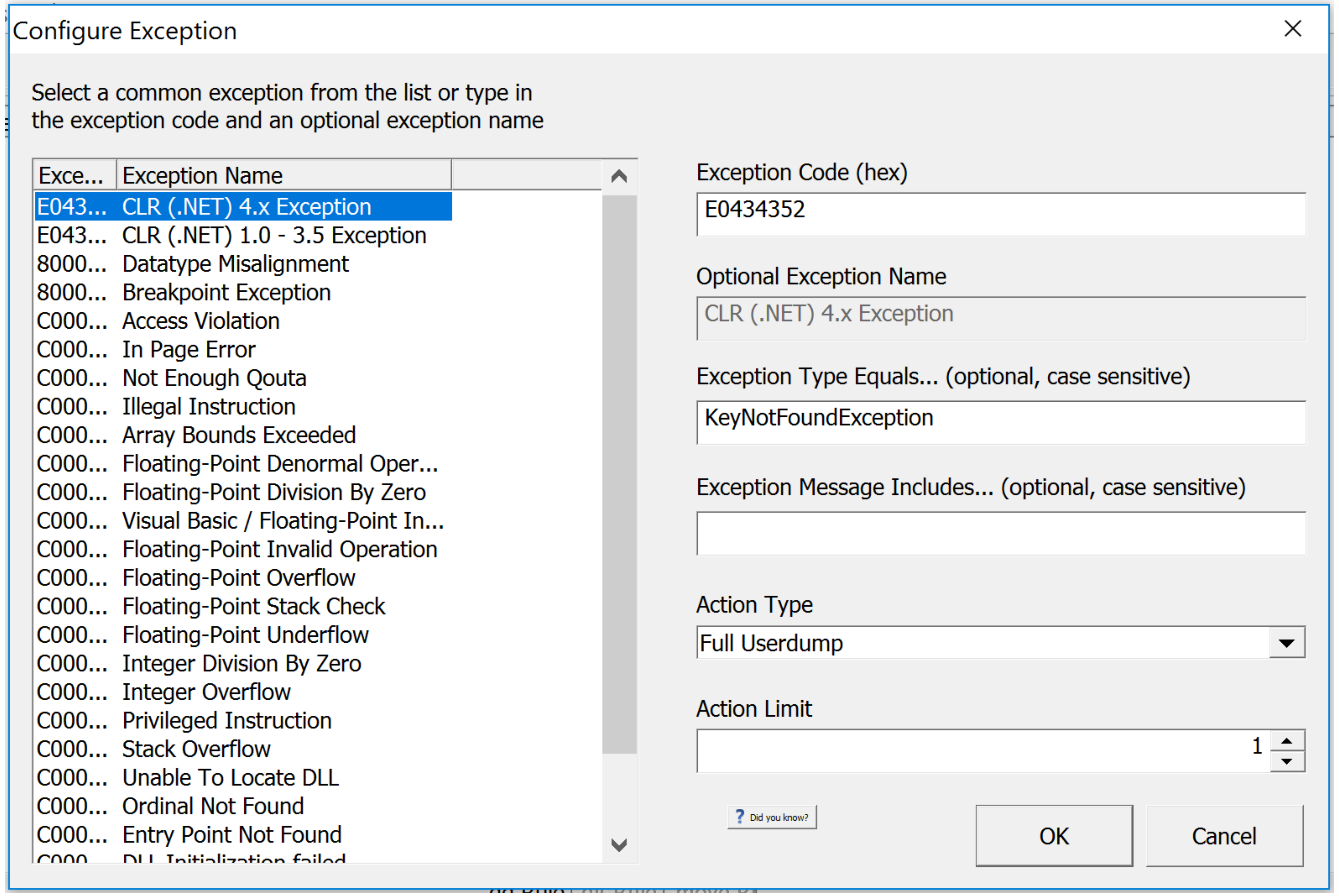
Capture user dumps used to troubleshoot performance problems including high CPU, deadlocks, long HTTP response times, and .NET Memory issues.

○ Native (non-.NET) Memory and Handle Leak

Use LeakTrack to track outstanding memory allocations and kernel object handles. Capture user dumps used to analyze memory and handle leaks.

□ Do not show this wizard automatically on startup.

DebugDiag - On Exceptions



Configure Exception

Select a common exception from the list or type in the exception code and an optional exception name

Exce...	Exception Name
E043...	CLR (.NET) 4.x Exception
E043...	CLR (.NET) 1.0 - 3.5 Exception
8000...	Datatype Misalignment
8000...	Breakpoint Exception
C000...	Access Violation
C000...	In Page Error
C000...	Not Enough Qouta
C000...	Illegal Instruction
C000...	Array Bounds Exceeded
C000...	Floating-Point Denormal Oper...
C000...	Floating-Point Division By Zero
C000...	Visual Basic / Floating-Point In...
C000...	Floating-Point Invalid Operation
C000...	Floating-Point Overflow
C000...	Floating-Point Stack Check
C000...	Floating-Point Underflow
C000...	Integer Division By Zero
C000...	Integer Overflow
C000...	Privileged Instruction
C000...	Stack Overflow
C000...	Unable To Locate DLL
C000...	Ordinal Not Found
C000...	Entry Point Not Found
C000...	DLL Initialization failed

Exception Code (hex): E0434352

Optional Exception Name: CLR (.NET) 4.x Exception

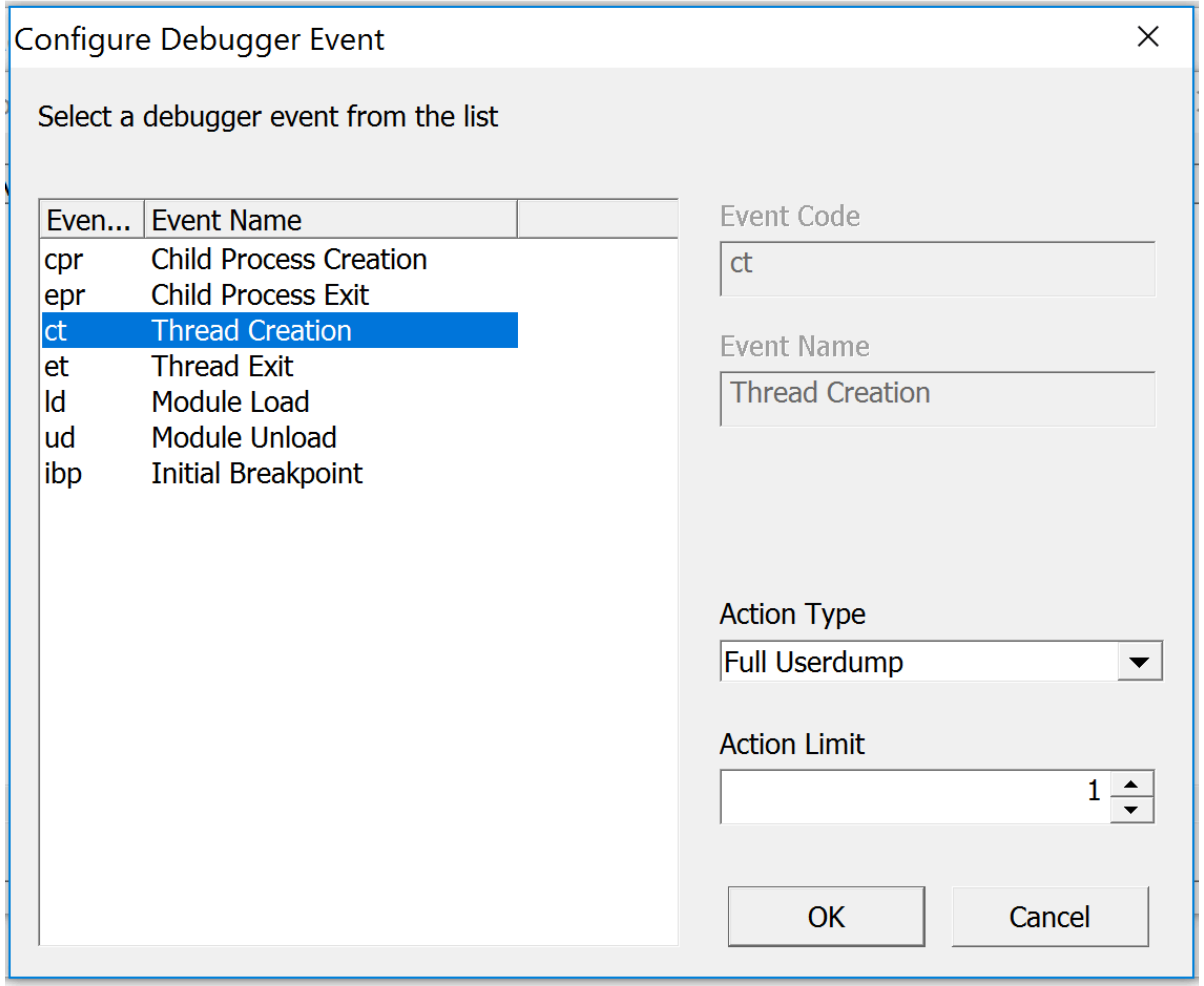
Exception Type Equals... (optional, case sensitive): KeyNotFoundException

Exception Message Includes... (optional, case sensitive):

Action Type: Full Userdump

Action Limit: 1

DebugDiag - On Events



Select a debugger event from the list

Even...	Event Name
cpr	Child Process Creation
epr	Child Process Exit
ct	Thread Creation
et	Thread Exit
ld	Module Load
ud	Module Unload
ibp	Initial Breakpoint

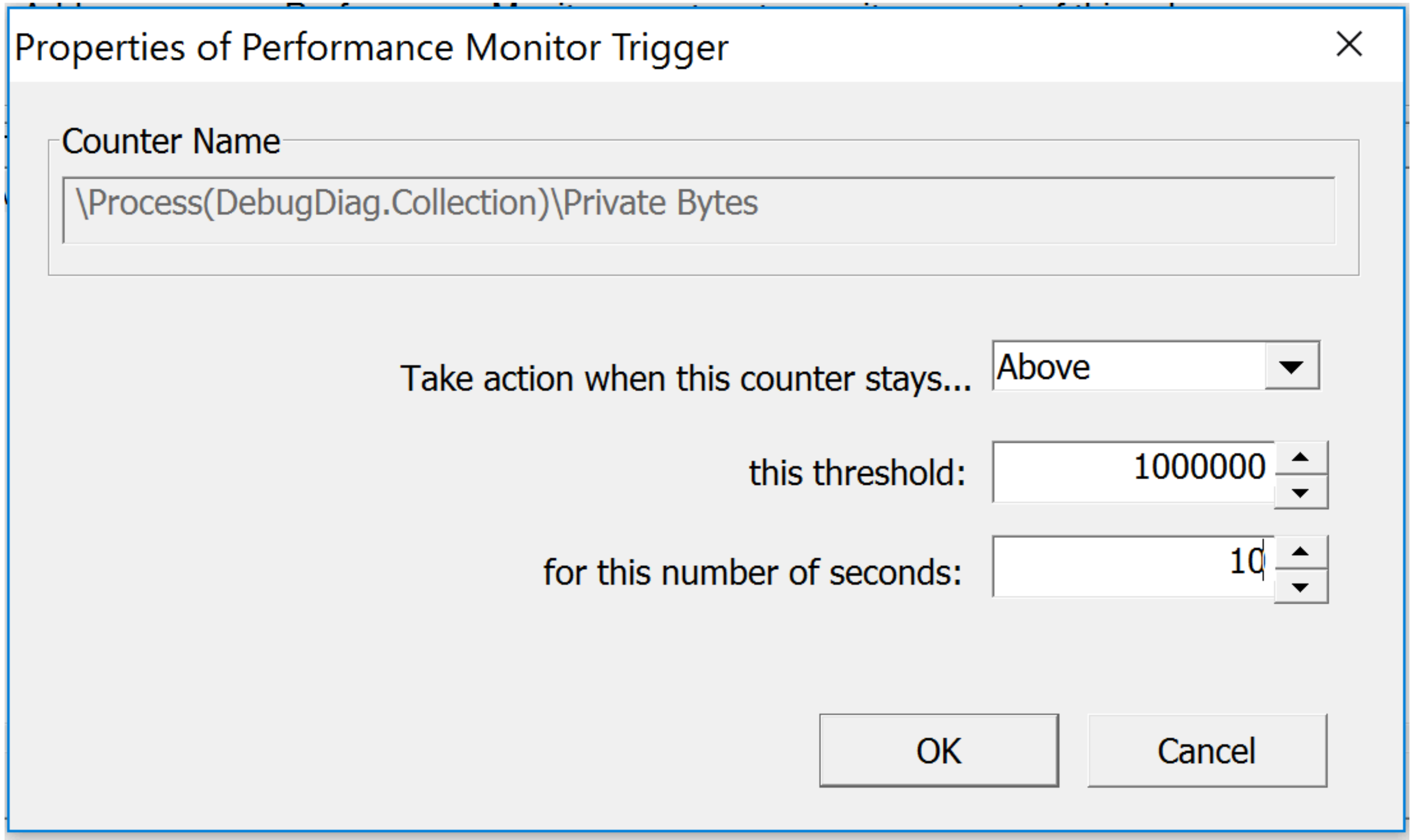
Event Code: ct

Event Name: Thread Creation

Action Type: Full Userdump

Action Limit: 1

DebugDiag - On a Perf Trigger



Properties of Performance Monitor Trigger

Counter Name

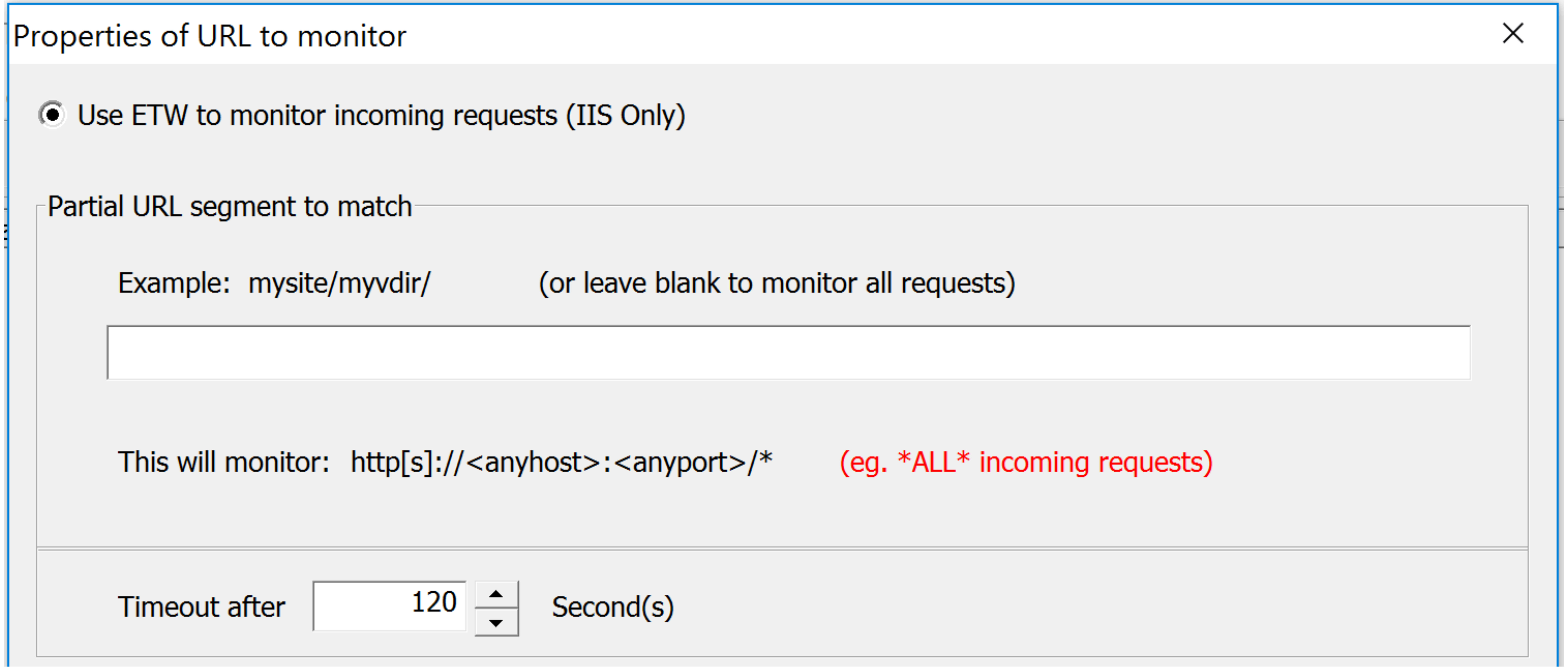
\Process(DebugDiag.Collection)\Private Bytes

Take action when this counter stays... Above

this threshold: 1000000

for this number of seconds: 10

DebugDiag - For IIS



Properties of URL to monitor

● Use ETW to monitor incoming requests (IIS Only)

Partial URL segment to match

Example: mysite/myvdir/ (or leave blank to monitor all requests)

This will monitor: http[s]://<anyhost>:<anyport>/* (eg. *ALL* incoming requests)

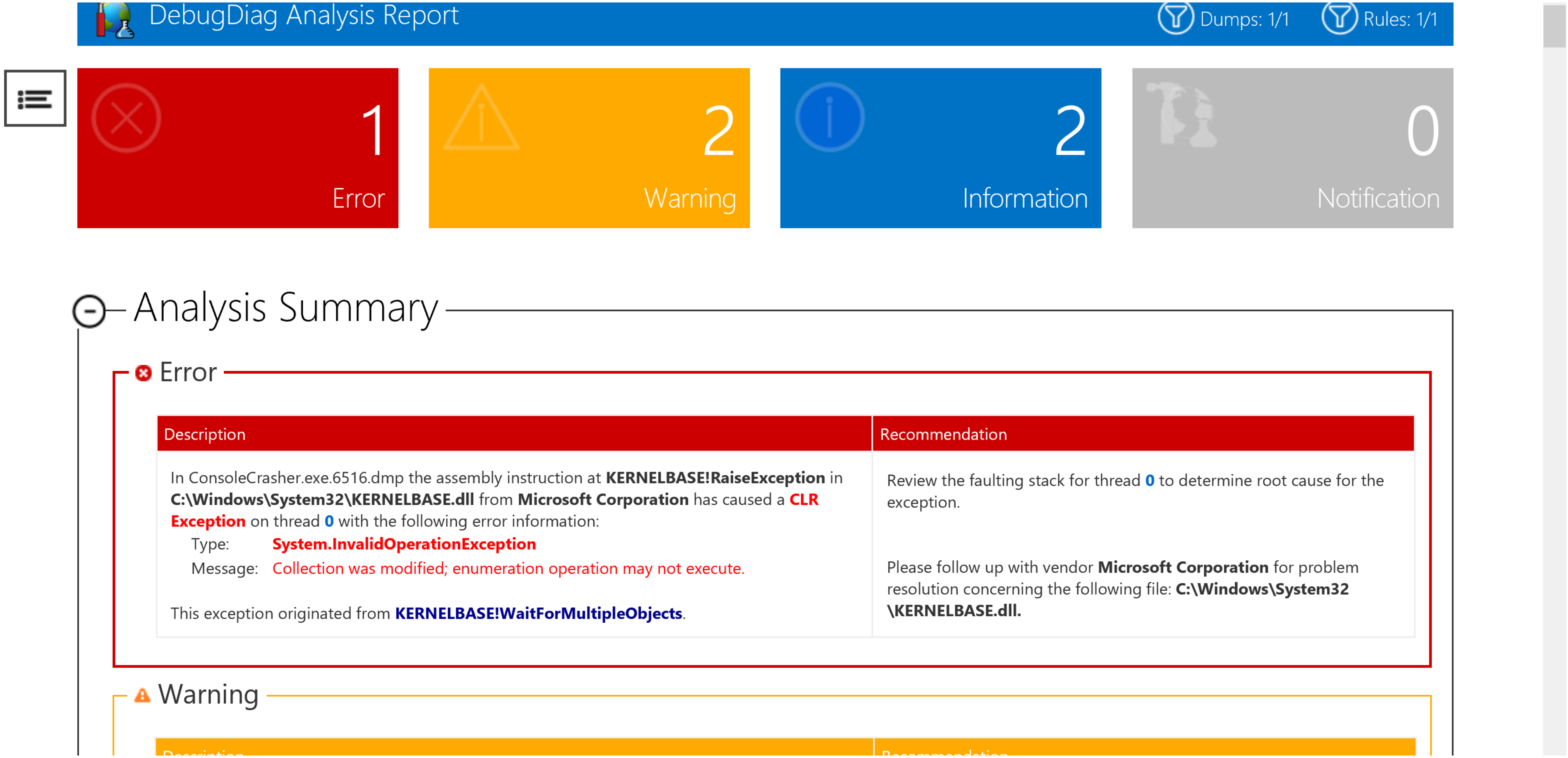
Timeout after 120 Second(s)

Linux

DEMO

Manual Dump Analysis

DebugDiag to the Rescue



Analysis Summary

Error

Description	Recommendation
In ConsoleCrasher.exe.6516.dmp the assembly instruction at **KERNELBASE!RaiseException** in **C:\Windows\System32\KERNELBASE.dll** from **Microsoft Corporation** has caused a **CLR Exception** on thread **0** with the following error information: Type: **System.InvalidOperationException** Message: Collection was modified; enumeration operation may not execute. This exception originated from **KERNELBASE!WaitForMultipleObjects**.	Review the faulting stack for thread **0** to determine root cause for the exception. Please follow up with vendor **Microsoft Corporation** for problem resolution concerning the following file: **C:\Windows\System32\KERNELBASE.dll.**

Warning

WinDbg Basic Analysis

DEMO

WinDbg Extensions

- It’s possible to write extension DLLs for WinDbg with your own customized behavior
- The SOS extension (part of .NET distribution) provides managed debugging functionality to WinDbg
  - Heap traversal, GC roots, type histograms
  - Managed threads and callstacks, managed exceptions

WinDbg Extensions

- It’s possible to write extension DLLs for WinDbg with your own customized behavior

- The SOS extension (part of .NET distribution) provides managed debugging functionality to WinDbg

    - Heap traversal, GC roots, type histograms

    - Managed threads and callstacks, managed exceptions

    - **Important: SOS version must match CLR version**

WinDbg Extensions

- It’s possible to write extension DLLs for WinDbg with your own customized behavior
- The SOS extension (part of .NET distribution) provides managed debugging functionality to WinDbg
  - Heap traversal, GC roots, type histograms
  - Managed threads and callstacks, managed exceptions
  - **Important: SOS version must match CLR version**
- More useful extensions: SOSEX, NetExt

Deadlocks with SOSEX

DEMO

Heap Walk with LLDB

DEMO

Automated Dump Analysis

Automation Approaches

- WinDbg/LLDB automation
    - For each dump:
        - Launch the debugger with initial commands, pipe to a file, parse the file

Automation Approaches

- WinDbg/LLDB automation
  - For each dump:
    - Launch the debugger with initial commands, pipe to a file, parse the file
- ClrMD
  - .NET library for dump analysis and live process inspection
  - Object model around: threads, heaps, exceptions, types, etc.
  - Windows-only for now

Automate WinDbg

DEMO

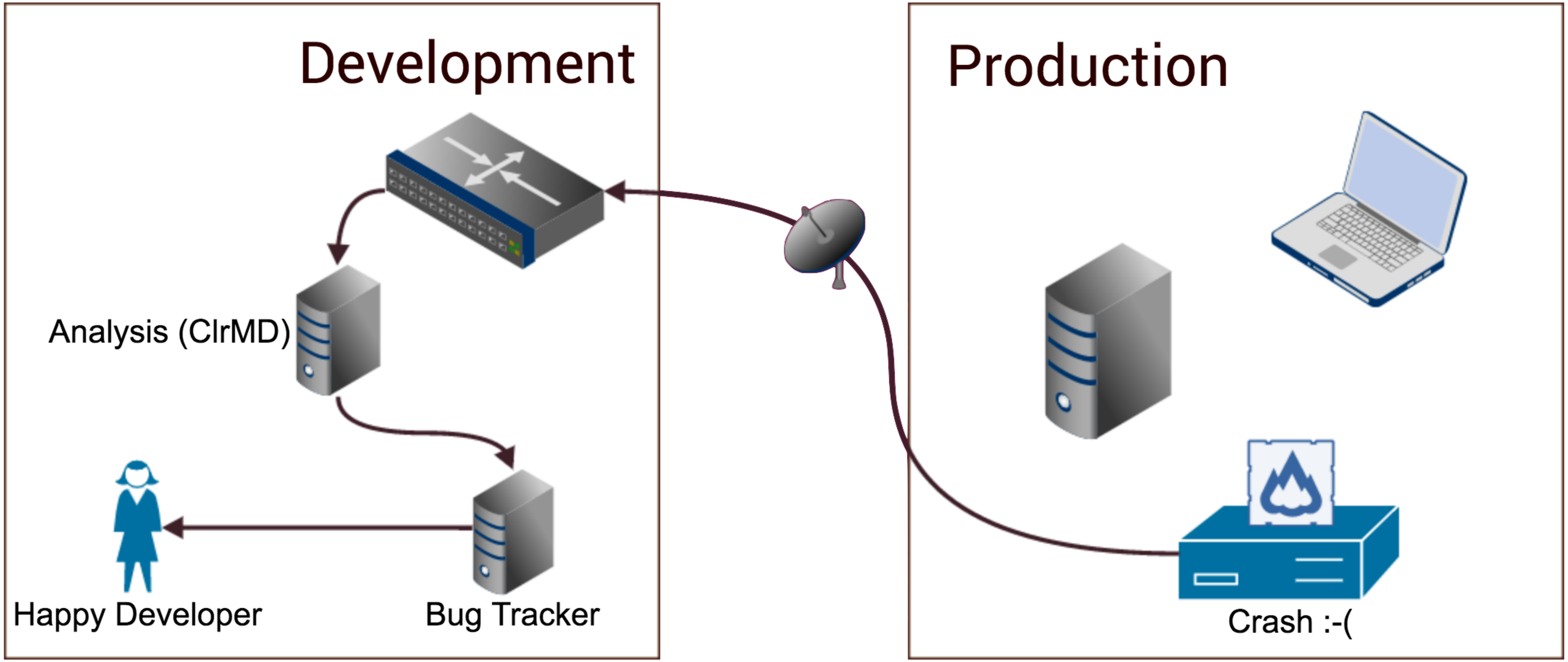
LLDB Scripts

DEMO

ClrMD

DEMO

What’s Next?


Development
Analysis (ClrMD)
Happy Developer
Bug Tracker
Production
Crash :-(

More Tools Available

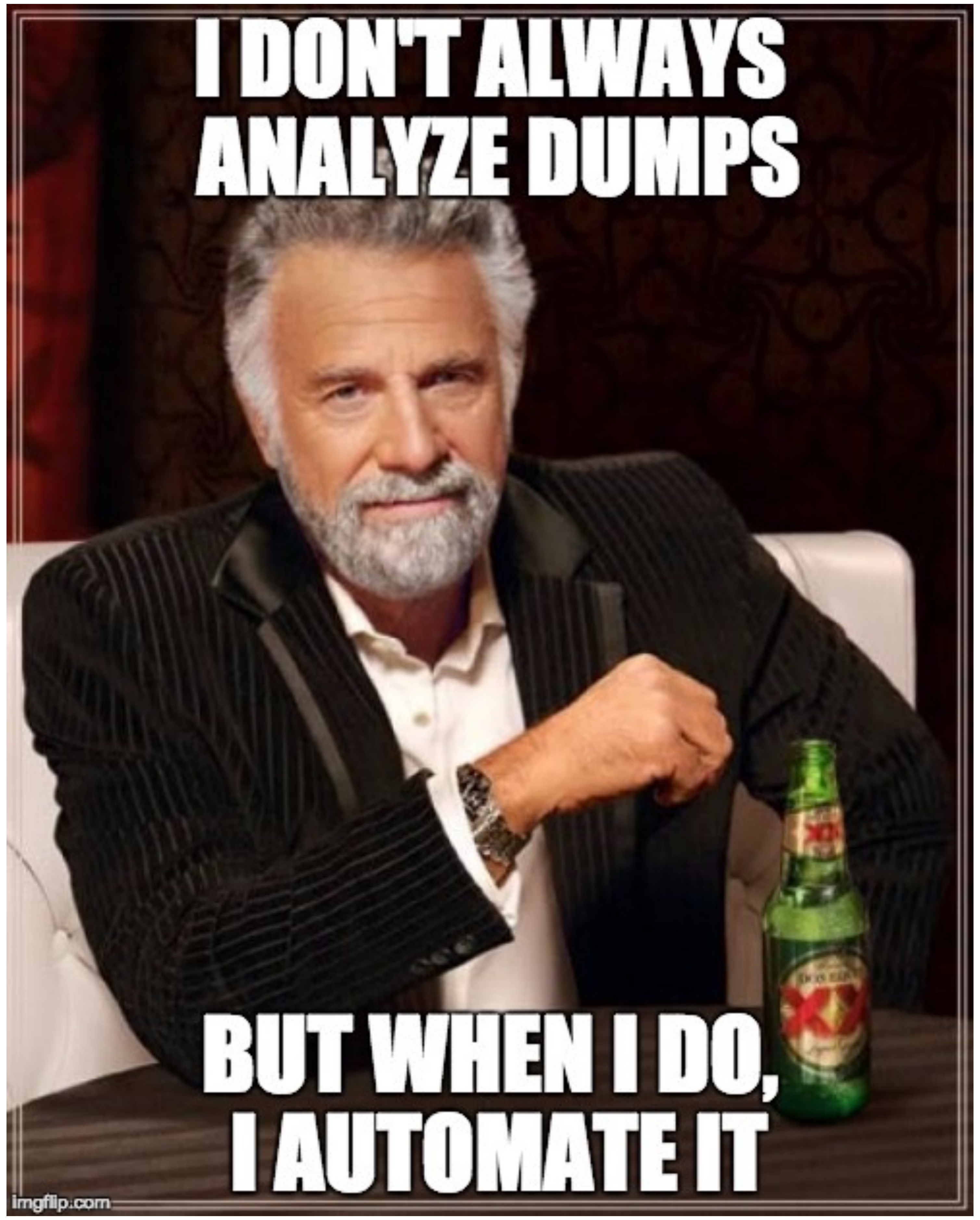
- msos - an open-source command-line tool based on ClrMD for Windows dump and live process analysis

- SuperDump - an open-source Windows dump analysis service allowing uploading a dump and getting a basic analysis

- Dr. Dump - commercial dump analysis service (under development)

Summary

KEEP
CALM
AND
TAKE A
DUMP


I DON'T ALWAYS
ANALYZE DUMPS
BUT WHEN I DO,
I AUTOMATE IT
imgflip.com

DOTNEXT

Thank You

Dina Goldshtein
Riverbed

dinazil@gmail.com
@dinagozil

riverbed®