LiquidHub

consulting | solutions | outsourcing

An introduction to production debugging

Brian Lyttle

Help us make this even better!

- Thank our sponsors!
  - This wouldn’t be possible without them.
- Please complete an online evaluation!

About me

- Brian Lyttle
  - Solution Architect at LiquidHub
- E-mail
  - blyttle@liquidhub.com
- Interwebs
  - @brianly on Twitter
  - @brianly on Stack Overflow
  - Blog infrequently at http://brianlyttle.com

Agenda

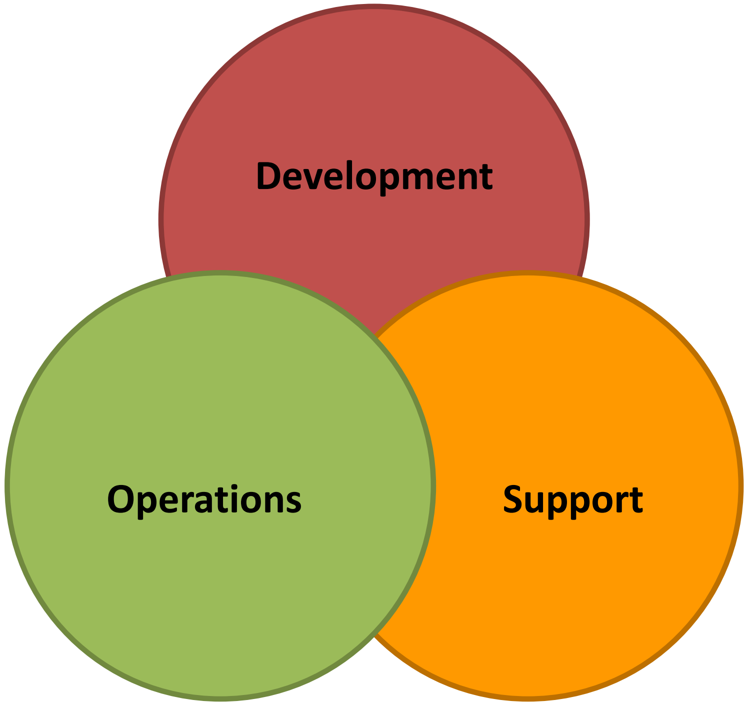
- What is production debugging?
- Debugging fundamentals
- Monitoring and instrumentation
- Capturing crash information
- Common scenarios
- … all with a healthy number of examples

Since this is an introductory talk I’m skipping:

  – WinDbg, SOS, and ADPlus
  – Native debugging

Audience

- Production debugging is something that helps a lot of different people
- A lot of these tools and techniques are not known by many developers.
- If you ever get the opportunity, go to a Mark Russinovich talk. The IT guys will know of him.



What is production debugging?

- Effectively any debugging you do once code has left the confines of a developer workstation.
- A variety of tools are required to investigate problems with:
  - Resource usage
  - Security or privilege violations
  - Random crashing
  - Bugs with 3rd party software
  - Network failures
- Some consider it a “black art” only accessible to the alpha developers amongst us.
- As a discipline it requires you to learn more about the internals of your operating system, database etc.

Why is this important?

- It will make your users and manager happy :)
- You will get home on time.
- Your relationship with operations will improve.
- As you do more of this, your knowledge of operating systems, databases etc. will improve.
- Eventually your thinking will change, and you become more proactive.

Production debugging techniques

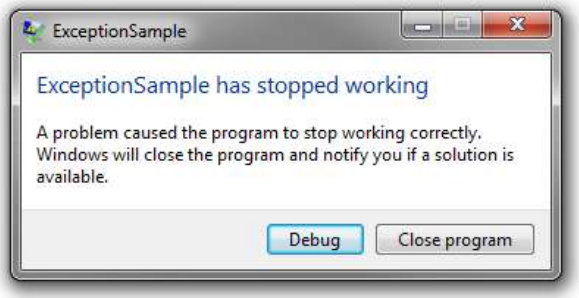
- Often it does not involve Visual Studio, or any other development environment.
- Active methods
  - Production debuggers
  - Tracing
  - Protocol analysis
- Passive methods
  - User reports
  - Crash dumps
  - Log files

Intermediate language and metadata

- Your .NET (managed code) assemblies are compiled to IL which is similar to assembly language.
- Older platforms like C/C++, classic VB, and many other do not have the concept of IL or metadata.
- With managed code an executable contains IL plus type information (metadata).
- This metadata is useful for Intellisense, serializing objects, debugging, and many other things.
- An interesting property of metadata is that it can help with disassembling your .NET assembly back to C#.

Exceptions decomposed

- Errors are supposed to be exceptional events within an application.
- So we have the **System.Exception** for conveying information about these errors.
- Provides a lot of important information:
  - Stack trace
  - Target site
  - Custom message
  - Inner exceptions (if any)


ExceptionSample

ExceptionSample has stopped working

A problem caused the program to stop working correctly.
Windows will close the program and notify you if a solution is available.

Obfuscating exceptions

- Each exception provides evidence for the person debugging the code.
- Like any type of evidence it can be tainted.
- Only catch and work with exceptions you can handle.
  - Don't default to catching System.Exception.
- If you catch an exception and need to re-throw it, make sure to throw the original.

Good

```
} catch (ArgumentException ex) {
   Log(ex.Message);
   throw;
}
```

1. Logs the exception.
2. Throws the original exception, not a new one.

Bad

```
} catch (Exception ex) {
   Log(ex.Message);
   throw ex;
}
```

```
} catch (Exception e) {
   Log(ex.Message);
   throw new Exception("", ex);
}
```

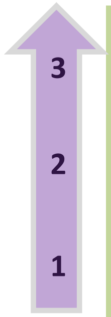
Debugging symbols

- You might have noticed that there are debug and release options in Visual Studio.
- If you build in **debug mode** then additional code to support the debugger, and symbol (.PDB) files are generated.
- In **release mode** an optimized version of your application is built. This might still create .PDB files.
- PDB files are called symbols. They map your executable code to the matching Visual Studio project.
  - Example: line number information.
- Best practice - keep a copy of the PDB files for every build you release.

Reading a stack trace

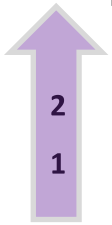
- Reading stack traces is a core skill for .NET developers.
- The lines shown below provide insight into what was happening.

Debug Build



```
Unhandled Exception: System.DivideByZeroException: Attempted to divide by zero.
   at CCExample1.Program.Divide(Int32 op1, Int32 op2) in C:\Code Camp\Projects\C
CExample1\CCExample1\Program.cs:line 65
   at CCExample1.Program.RunCode() in C:\Code Camp\Projects\CCExample1\CCExample
1\Program.cs:line 36
   at CCExample1.Program.Main(String[] args) in C:\Code Camp\Projects\CCExample1
\CCExample1\Program.cs:line 14
```

Release Build



```
Unhandled Exception: System.DivideByZeroException: Attempted to divide by zero.
   at CCExample1.Program.RunCode()
   at CCExample1.Program.Main(String[] args)
```

Log files

- Applications often have official and unofficial logs which you can mine for information.
- Windows examples:
  - Event Viewer
  - %windir%\ System32\LogFiles
  - %windir%\system32\wbem\logs
  - %COMMONPROGRAMFILES%\Microsoft Shared\web server extensions\12\LOGS
  - %systemdrive%\PerfLogs\
  - Temp directories (everyone can write there!)
- Applications might also log to their database, or to a directory under their user identity.
  - Eg. C:\Users\brianly\AppData\Local\Diagnostics
- On Linux look in /var/log.

System resources

It is very important to understand how resources constrain your application. They don’t seem that important until you run out of them.

At times they will seem plentiful, but that may not be the case under the covers.

Most common

- Memory
- CPU
- Disk I/O
- Network I/O

Meta resources

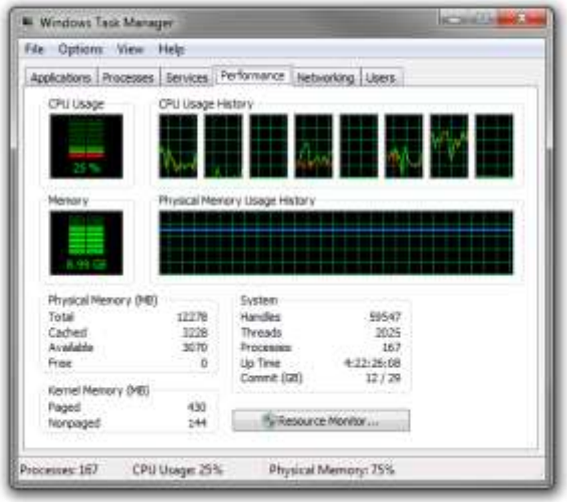
- Handles
- Security privileges
- Application caches
- Software licenses
- Garbage collections
- Complex algorithms

Remote debugging

- For some people it might be possible to use the Visual Studio remote debugging tools.
- These let you debug applications across the network with a familiar experience.
- For a lot of people this is just not possible:
  - Security considerations
  - Firewalls and routing
  - Resistance from administrator
  - End user impact

Enter Task Manager

- Available on any Windows computer.
- Brent Ozar: a “check engine light” for your system.
- You get a high level view but a lot of interesting information is hidden by default.
- Special processes are not visible.
- Deeper info on Threads or other esoteric objects is not available.
- Resource Monitor is a useful partner for Task Manager on Vista/7.
- Fact: On Windows 2000 and earlier it can be plain wrong.

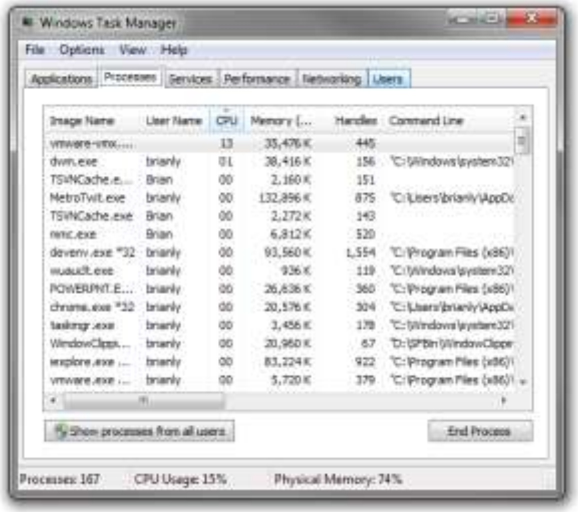


Physical Memory (MB)	
Total	12278
Cached	3228
Available	3070
Free	0

Kernel Memory (MB)	
Paged	430
Nonpaged	144

System	
Handles	59547
Threads	2025
Processes	167
Up Time	4:22:26:08
Commit (GB)	12 / 29

Processes: 167 CPU Usage: 25% Physical Memory: 75%



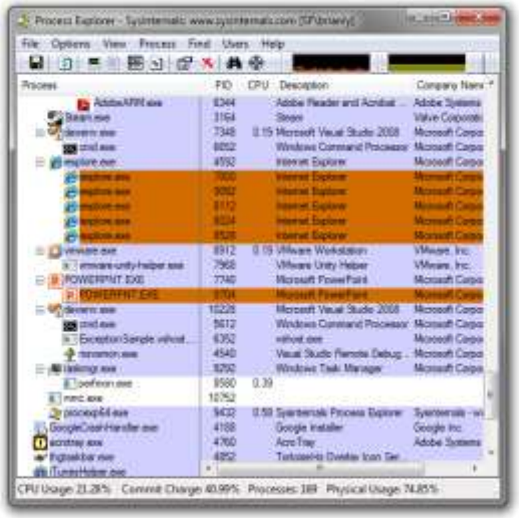
Windows Task Manager
File Options View Help
Applications Processes Services Performance Networking Users
Show processes from all users
End Process

Image Name	User Name	CPU	Memory (...	Handles	Command Line
vmware-vmx....		13	35,476 K	445	
dwm.exe	brianly	01	38,416 K	156	"C:\Windows\system32\
TSVNCache.e...	Brian	00	2,160 K	151	
MetroTwit.exe	brianly	00	132,896 K	875	"C:\Users\brianly\AppD
TSVNCache.exe	Brian	00	2,272 K	143	
mmc.exe	Brian	00	6,812 K	520	
devenv.exe *32	brianly	00	93,560 K	1,554	"C:\Program Files (x86)\
wuauclt.exe	brianly	00	936 K	119	"C:\Windows\system32\
POWERPNT.E...	brianly	00	26,636 K	360	"C:\Program Files (x86)\
chrome.exe *32	brianly	00	20,576 K	304	"C:\Users\brianly\AppD
taskmgr.exe	brianly	00	3,456 K	178	"C:\Windows\system32\
WindowClipp...	brianly	00	20,960 K	67	"D:\SPBin\WindowClippe
iexplore.exe ...	brianly	00	83,224 K	922	"C:\Program Files (x86)\
vmware.exe ...	brianly	00	5,720 K	379	"C:\Program Files (x86)\

Processes: 167 CPU Usage: 15% Physical Memory: 74%

Process Explorer

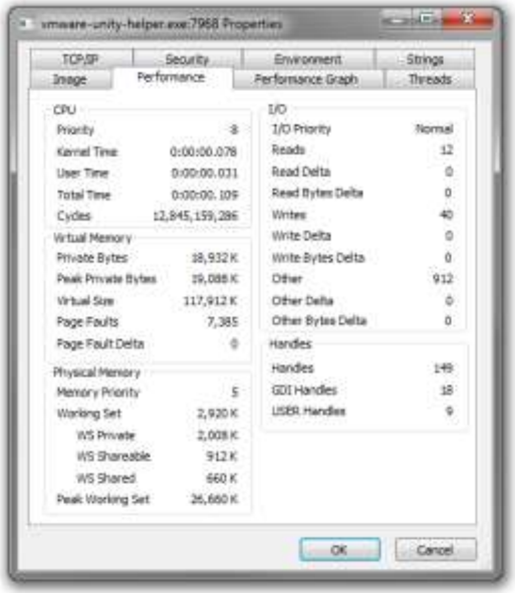
- Improves on the capabilities of Task Manager.
- Highlights Process activity.
- Hierarchical view.
- All the gritty details.
- See which individual Thread is doing all the work.


Process Explorer - Sysinternals: www.sysinternals.com [SF\brianv]

File Options View Process Find Users Help

Process	PID	CPU	Description	Company Name
AdobeARM.exe	6344		Adobe Reader and Acrobat ...	Adobe Systems
Steam.exe	3164		Steam	Valve Corporati
devenv.exe	7348	0.15	Microsoft Visual Studio 2008	Microsoft Corpo
cmd.exe	8052		Windows Command Processor	Microsoft Corpo
iexplore.exe	4592		Internet Explorer	Microsoft Corpo
iexplore.exe	7000		Internet Explorer	Microsoft Corpo
iexplore.exe	9092		Internet Explorer	Microsoft Corpo
iexplore.exe	8112		Internet Explorer	Microsoft Corpo
iexplore.exe	8324		Internet Explorer	Microsoft Corpo
iexplore.exe	8528		Internet Explorer	Microsoft Corpo
vmware.exe	8912	0.19	VMware Workstation	VMware, Inc.
vmware-unity-helper.exe	7968		VMware Unity Helper	VMware, Inc.
POWERPNT.EXE	7740		Microsoft PowerPoint	Microsoft Corpo
POWERPNT.EXE	8704		Microsoft PowerPoint	Microsoft Corpo
devenv.exe	10228		Microsoft Visual Studio 2008	Microsoft Corpo
cmd.exe	5612		Windows Command Processor	Microsoft Corpo
ExceptionSample.vshost...	6352		vshost.exe	Microsoft Corpo
msvsmon.exe	4540		Visual Studio Remote Debug...	Microsoft Corpo
taskmgr.exe	8292		Windows Task Manager	Microsoft Corpo
perfmon.exe	8580	0.39		
mmc.exe	10752			
procexp64.exe	9432	0.58	Sysinternals Process Explorer	Sysinternals - ww
GoogleCrashHandler.exe	4188		Google Installer	Google Inc.
acrotray.exe	4760		AcroTray	Adobe Systems
thgtaskbar.exe	4852		TortoiseHg Overlay Icon Ser...	
iTunesHelper.exe				

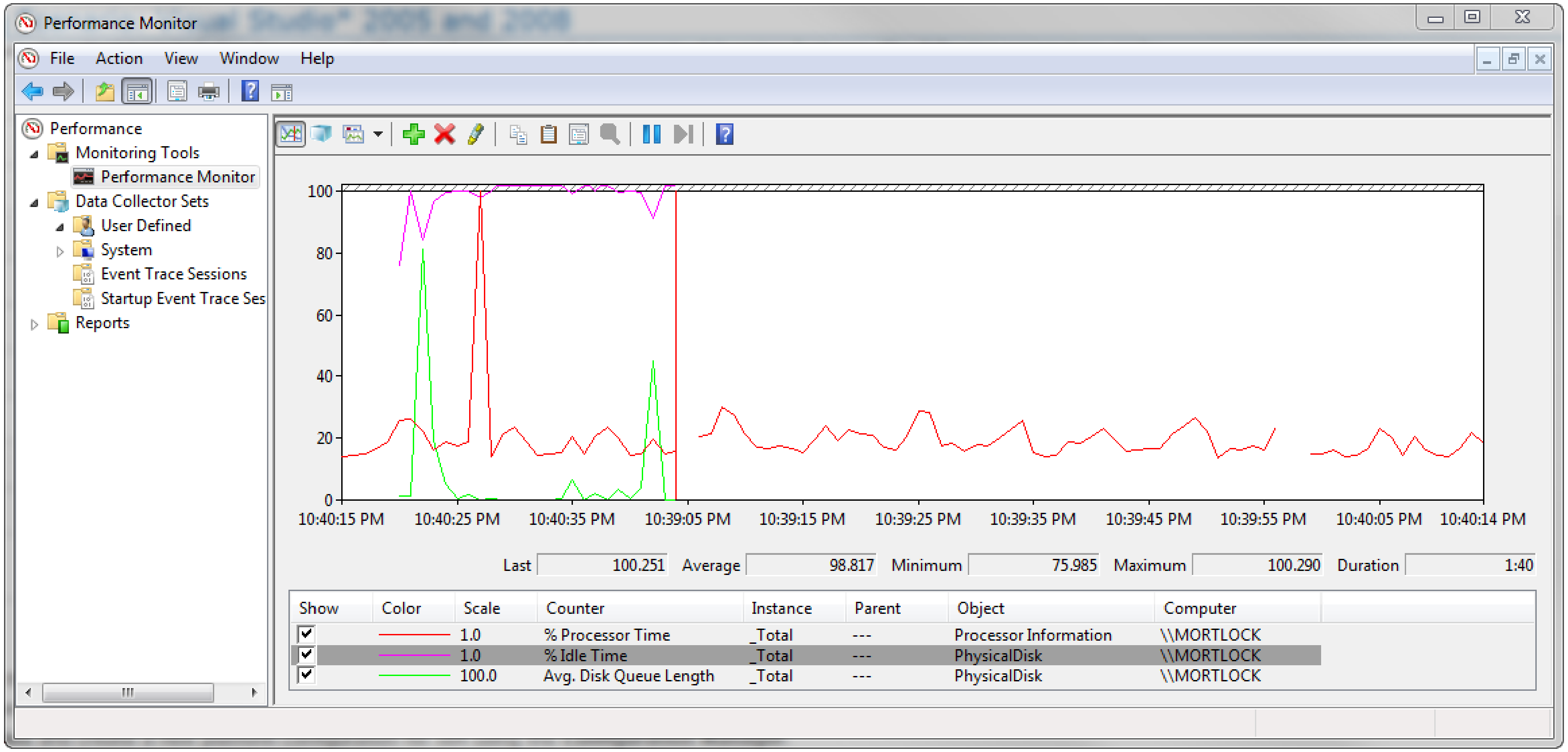
CPU Usage: 21.28% Commit Charge: 40.99% Processes: 169 Physical Usage: 74.85%


vmware-unity-helper.exe:7968 Properties

CPU	
Priority	8
Kernel Time	0:00:00.078
User Time	0:00:00.031
Total Time	0:00:00.109
Cycles	12,845,159,286
Virtual Memory	
Private Bytes	18,932 K
Peak Private Bytes	19,088 K
Virtual Size	117,912 K
Page Faults	7,385
Page Fault Delta	0
Physical Memory	
Memory Priority	5
Working Set	2,920 K
WS Private	2,008 K
WS Shareable	912 K
WS Shared	660 K
Peak Working Set	26,660 K
I/O	
I/O Priority	Normal
Reads	12
Read Delta	0
Read Bytes Delta	0
Writes	40
Write Delta	0
Write Bytes Delta	0
Other	912
Other Delta	0
Other Bytes Delta	0
Handles	
Handles	149
GDI Handles	18
USER Handles	9

Performance Monitor

Process Explorer is good for spot checks, but you need something better to do background monitoring.



Last	Average	Minimum	Maximum	Duration
100.251	98.817	75.985	100.290	1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
✓	🟥	1.0	% Processor Time	_Total	---	Processor Information	\\MORTLOCK
✓	🟪	1.0	% Idle Time	_Total	---	PhysicalDisk	\\MORTLOCK
✓	🟩	100.0	Avg. Disk Queue Length	_Total	---	PhysicalDisk	\\MORTLOCK

Metrics Available

- Performance Monitor data can be captured for later review with a Data Collector Set.
- Export data to Excel for later analysis.
- It is possible to create custom performance counters.
- Examples:
  - User Login
  - Cart empties
  - Document uploads

CPU

- % Idle Time
- % User Time

Memory

- Available Mbytes
- Paging File % Usage

Disk

- Disk Reads/sec
- Disk Writes/sec
- Avg. Disk Queue Length

Others

- % Time in GC
- ASP.NET Cache Total Misses
- Windows Workflows Executing
- Faxes Sent ;)

Utilities

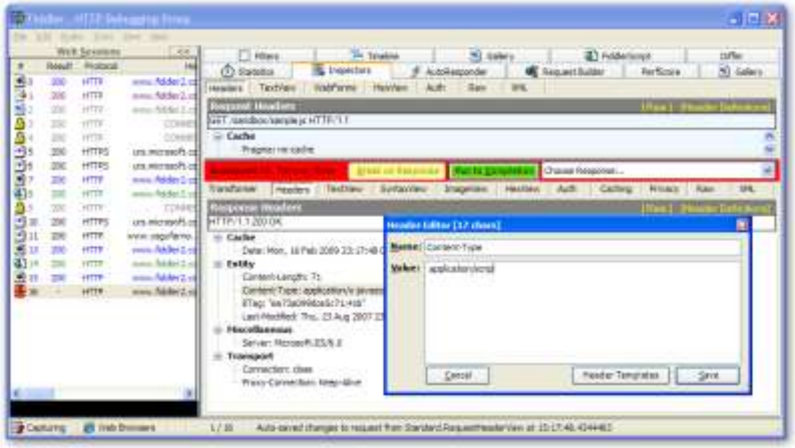
Name	Description
DebugView *	Displays debug messages on local and remote computers.
Process Monitor *	Captures Process, Registry, and File activity in a single utility. Replaces the famous Filemon and Regmon tools.
AccessEnum	Review file system and Registry permissions.
AccessChk	Command line tool similar to AccessEnum.
AD Insight	Troubleshoot and monitor client LDAP activity.
KerbTray *	View Kerberos security tickets.
Reflector *	.NET assembly decompiler and analyzer.
WPF Snoop	Snoop is a tools for exploring a running WPF application.
SysExporter *	Allows you to grab information from list boxes that don't provide a save/export option.
WinSpy++	Similar to WPF Snoop but for non-WPF applications.
Silverlight Spy	WPF Snoop-type tool for Silverlight.
Visual Log Parser	Front end for the great Log Parser utility from MS. Write SQL-like queries against raw log files. Much easier than messing with Perl or some other scripting utility.

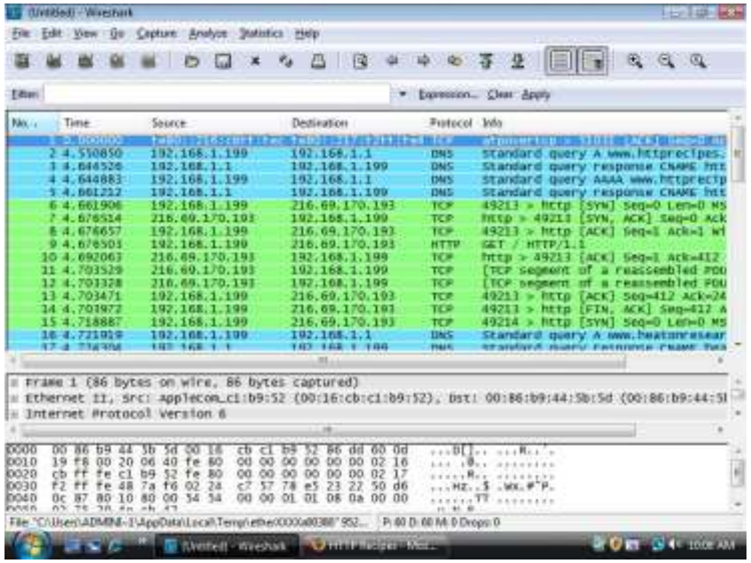
* Used in the past week

Protocol analysis

- Snoop on what the server or client are sending over the network.
- These tools should be used with care. It is possible to capture passwords and other sensitive data. Work with a network admin when you need to perform a capture in a corporate environment.
- It is easiest to debug HTTP services with one of the specialized tools.

Protocol	Tools
HTTP(S)	Fiddler, Charles Proxy, HttpWatch
Others	Network Monitor, Wireshark





```
No.  Time      Source           Destination      Protocol Info
  2 4.550850  192.168.1.199    192.168.1.1      DNS      Standard query A www.httprecipes.
  3 4.644526  192.168.1.1      192.168.1.199    DNS      Standard query response CNAME htt
  4 4.644883  192.168.1.199    192.168.1.1      DNS      Standard query AAAA www.httprecip
  5 4.661212  192.168.1.1      192.168.1.199    DNS      Standard query response CNAME htt
  6 4.661906  192.168.1.199    216.69.170.193   TCP      49213 > http [SYN] Seq=0 Len=0 MS
  7 4.676514  216.69.170.193   192.168.1.199    TCP      http > 49213 [SYN, ACK] Seq=0 Ack
  8 4.676657  192.168.1.199    216.69.170.193   TCP      49213 > http [ACK] Seq=1 Ack=1 Wi
  9 4.676501  192.168.1.199    216.69.170.193   HTTP     GET / HTTP/1.1
 10 4.692063  216.69.170.193   192.168.1.199    TCP      http > 49213 [ACK] Seq=1 Ack=412
 11 4.703529  216.69.170.193   192.168.1.199    TCP      [TCP segment of a reassembled PDU
 12 4.701328  216.69.170.193   192.168.1.199    TCP      [TCP segment of a reassembled PDU
 13 4.703471  192.168.1.199    216.69.170.193   TCP      49213 > http [ACK] Seq=412 Ack=24
 14 4.703972  192.168.1.199    216.69.170.193   TCP      49213 > http [FIN, ACK] Seq=412 A
 15 4.718887  192.168.1.199    216.69.170.193   TCP      49214 > http [SYN] Seq=0 Len=0 MS
 16 4.721919  192.168.1.199    192.168.1.1      DNS      Standard query A www.heatonresear

Frame 1 (86 bytes on wire, 86 bytes captured)
Ethernet II, Src: Applecom_c1:b9:52 (00:16:cb:c1:b9:52), Dst: 00:86:b9:44:5b:5d (00:86:b9:44:5b
Internet Protocol Version 6

0000  00 86 b9 44 5b 5d 00 16  cb c1 b9 52 86 dd 60 0d   ...D[].. ...R..`.
0010  19 f8 00 20 06 40 fe 80  00 00 00 00 00 00 02 16   ... .@.. ........
0020  cb ff fe c1 b9 52 fe 80  00 00 00 00 00 00 02 17   .....R.. ........
0030  f2 ff fe 48 7a f6 02 24  c7 57 78 e5 23 22 50 d6   ...Hz..$ .Wx.#"P.
0040  0c 87 80 10 80 00 54 54  00 00 01 01 08 0a 00 00   ......TT ........
```

What is a crash dump?

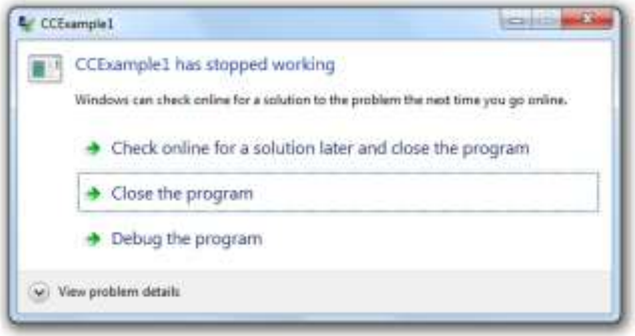
- These are basically files containing information on the exception which caused the crash.
- Unhandled exceptions bubble up to the top-level of an application.
- Then .NET invokes it’s own default error handling.
- Luckily you can override this behavior and do some interesting things.

Why generate crash dumps?

- Often users only know part of the story
  - They don’t know how the application works
  - Only a hard crash gets their full attention
  - Small details get ignored along the way
- Good for fixing “broken windows”
  - Management won’t let you work on the bits of an application that need some attention
  - You need some metrics to take to your manager

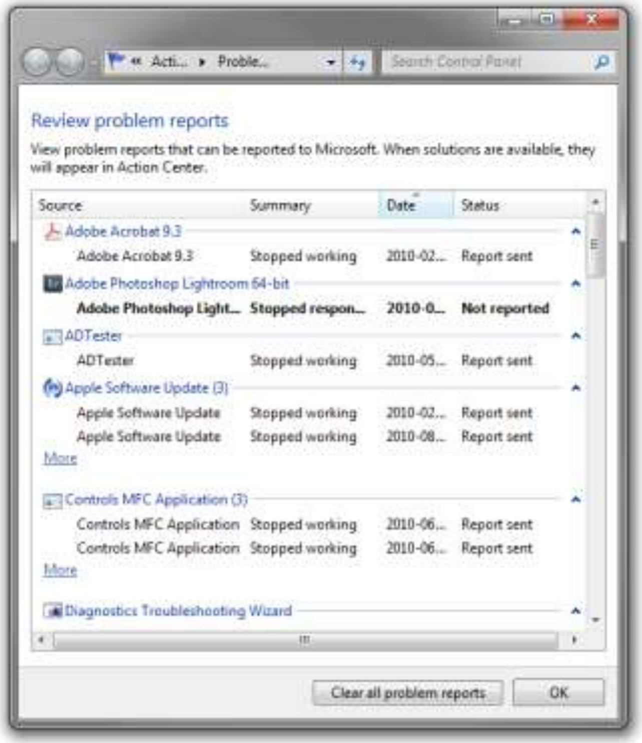
Windows Error Reporting

- Microsoft had a problem with buggy drivers making Windows bad.
- They needed to identify and fix this problem.
- End users weren't much help so a technical solution was needed.
- WER was born and Microsoft were able to identify and resolve the top bugs.



CCExample1 has stopped working

Windows can check online for a solution to the problem the next time you go online.



Review problem reports

View problem reports that can be reported to Microsoft. When solutions are available, they will appear in Action Center.

Source	Summary	Date	Status
Adobe Acrobat 9.3			
Adobe Acrobat 9.3	Stopped working	2010-02...	Report sent
Adobe Photoshop Lightroom 64-bit			
Adobe Photoshop Light...	**Stopped respon...**	**2010-0...**	**Not reported**
ADTester			
ADTester	Stopped working	2010-05...	Report sent
Apple Software Update (3)			
Apple Software Update	Stopped working	2010-02...	Report sent
Apple Software Update	Stopped working	2010-08...	Report sent
More			
Controls MFC Application (3)			
Controls MFC Application	Stopped working	2010-06...	Report sent
Controls MFC Application	Stopped working	2010-06...	Report sent
More			
Diagnostics Troubleshooting Wizard			

Crash dumps the hard way

- It is possible to write your own code to intercept an unhandled exception.
- Register for the event:

```
AppDomain.CurrentDomain.UnhandledException += new
UnhandledExceptionEventHandler(ExceptionHandler);
```

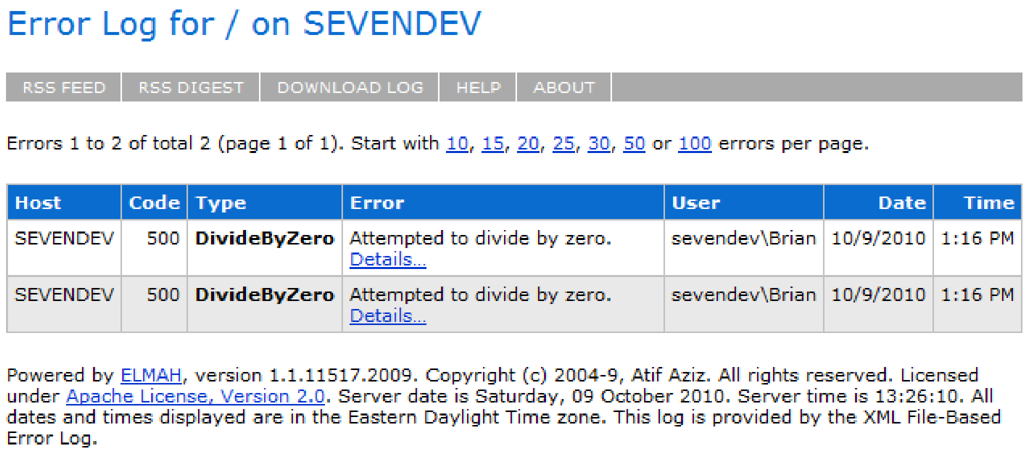
- Then write a single method to handle it:

```
void ExceptionHandler(object sender, UnhandledExceptionEventArgs e)
{
  // Do work...
  // Just make sure to quit!
}
```

- Just make sure to close the application, and write some code to upload the crash dump.
- It’s a good job someone has done this for us.

ELMAH makes it easy for ASP.NET

- ELMAH (Error Logging Modules and Handlers) is a free open source component.
- Features:
  - Log to many common databases.
  - Filter out certain error messages.
  - Supports a wide variety of hosting scenarios.
  - SharePoint supported in addition to Web Forms and ASP.NET MVC.
  - Admin web pages.
- Steps:
  1. Add sectionGroup.
  2. Add httpHandler.
  3. Add httpModule.
  4. Configure output options



Error Log for / on SEVENDEV

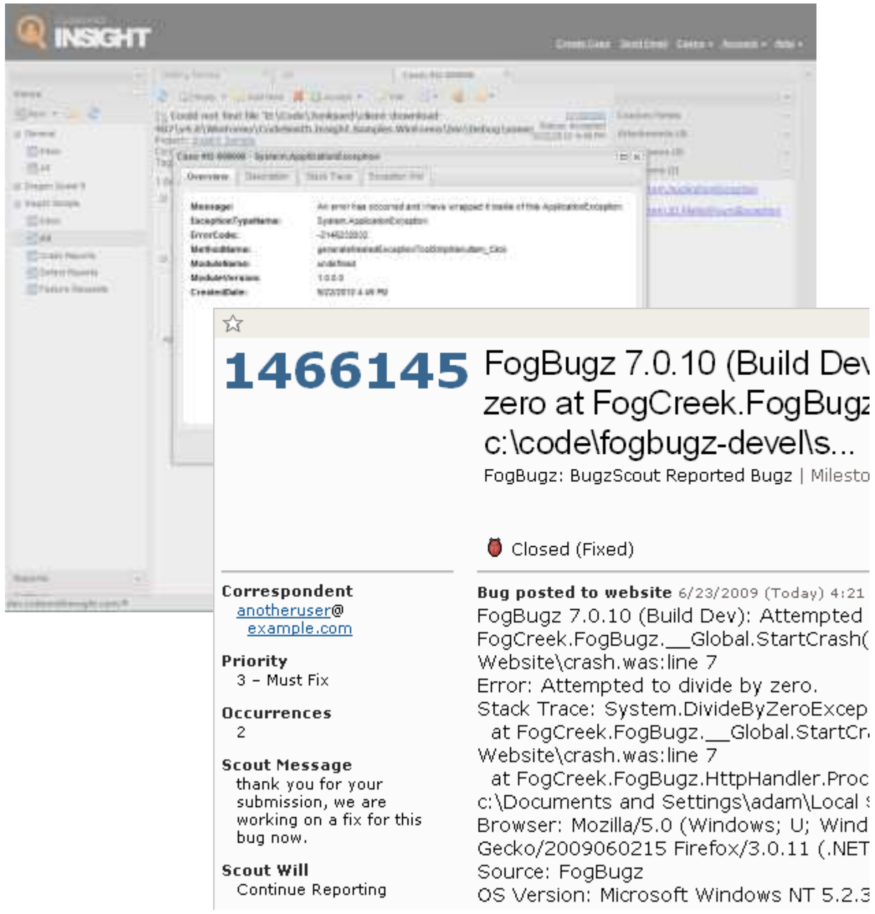
Errors 1 to 2 of total 2 (page 1 of 1). Start with 10, 15, 20, 25, 30, 50 or 100 errors per page.

Host	Code	Type	Error	User	Date	Time
SEVENDEV	500	**DivideByZero**	Attempted to divide by zero. Details…	sevendev\Brian	10/9/2010	1:16 PM
SEVENDEV	500	**DivideByZero**	Attempted to divide by zero. Details…	sevendev\Brian	10/9/2010	1:16 PM

Powered by ELMAH, version 1.1.11517.2009. Copyright (c) 2004-9, Atif Aziz. All rights reserved. Licensed under Apache License, Version 2.0. Server date is Saturday, 09 October 2010. Server time is 13:26:10. All dates and times displayed are in the Eastern Daylight Time zone. This log is provided by the XML File-Based Error Log.

Doing more with crash dumps

- Group by the Target Site and identify top exceptions.
- Dump the crash dump into your bug tracker and track the fix.
- FogBugz, CodeSmith Insight, Jira, and many other products have support.



1466145 FogBugz 7.0.10 (Build Dev
zero at FogCreek.FogBugz
c:\code\fogbugz-devel\s...
FogBugz: BugzScout Reported Bugz | Milesto

Closed (Fixed)

Correspondent
anotheruser@
example.com

Priority
3 – Must Fix

Occurrences
2

Scout Message
thank you for your submission, we are working on a fix for this bug now.

Scout Will
Continue Reporting

Bug posted to website 6/23/2009 (Today) 4:21
FogBugz 7.0.10 (Build Dev): Attempted
FogCreek.FogBugz.__Global.StartCrash(
Website\crash.was:line 7
Error: Attempted to divide by zero.
Stack Trace: System.DivideByZeroExcep
  at FogCreek.FogBugz.__Global.StartCr
Website\crash.was:line 7
  at FogCreek.FogBugz.HttpHandler.Proc
c:\Documents and Settings\adam\Local
Browser: Mozilla/5.0 (Windows; U; Wind
Gecko/2009060215 Firefox/3.0.11 (.NET
Source: FogBugz
OS Version: Microsoft Windows NT 5.2.3

References

- Books
  - Debugging Applications for Microsoft .NET 2.0 Windows
  - Advanced .NET Debugging
  - Advanced Windows Debugging
  - Debug It!
- Microsoft Documentation
  - Creating custom performance counters
  - Windows Event Tracing
  - Using Performance Monitor
  - Monitoring with SQL Profiler
  - Analyzing Network Data with Network Monitor
- Presentations
  - Windows Sysinternals Primer: Process Explorer, Process Monitor, and More

Thanks!

- Please complete an online evaluation!
- Prizes!