EKON 21

Creative Delphi Debugging Techniques

Brian Long

http://blong.com

http://blog.blong.com

Agenda

- Logging
- Debugging tricks and ‘gotchas’
- The CPU window
- Chasing down Access Violations
- Memory managers:
  - FastMM
  - FastMM4
  - SimpleHeap
- GFlags

Logging

- Evidence provision
- Execution flow tracing:
  - Commercial, e.g.
    - SmartInspect
    - Embarcadero (née Raize) CodeSite Studio 5
  - Open Source, e.g.
    - Overseer (old clone of CodeSite)
    - LoggerPro by Daniele Teti on github

Logging

- Execution flow tracing
  - Agricultural execution flow tracing
    - Message boxes – drawbacks...
  - Look Ma, no code!
    - Advanced breakpoint properties

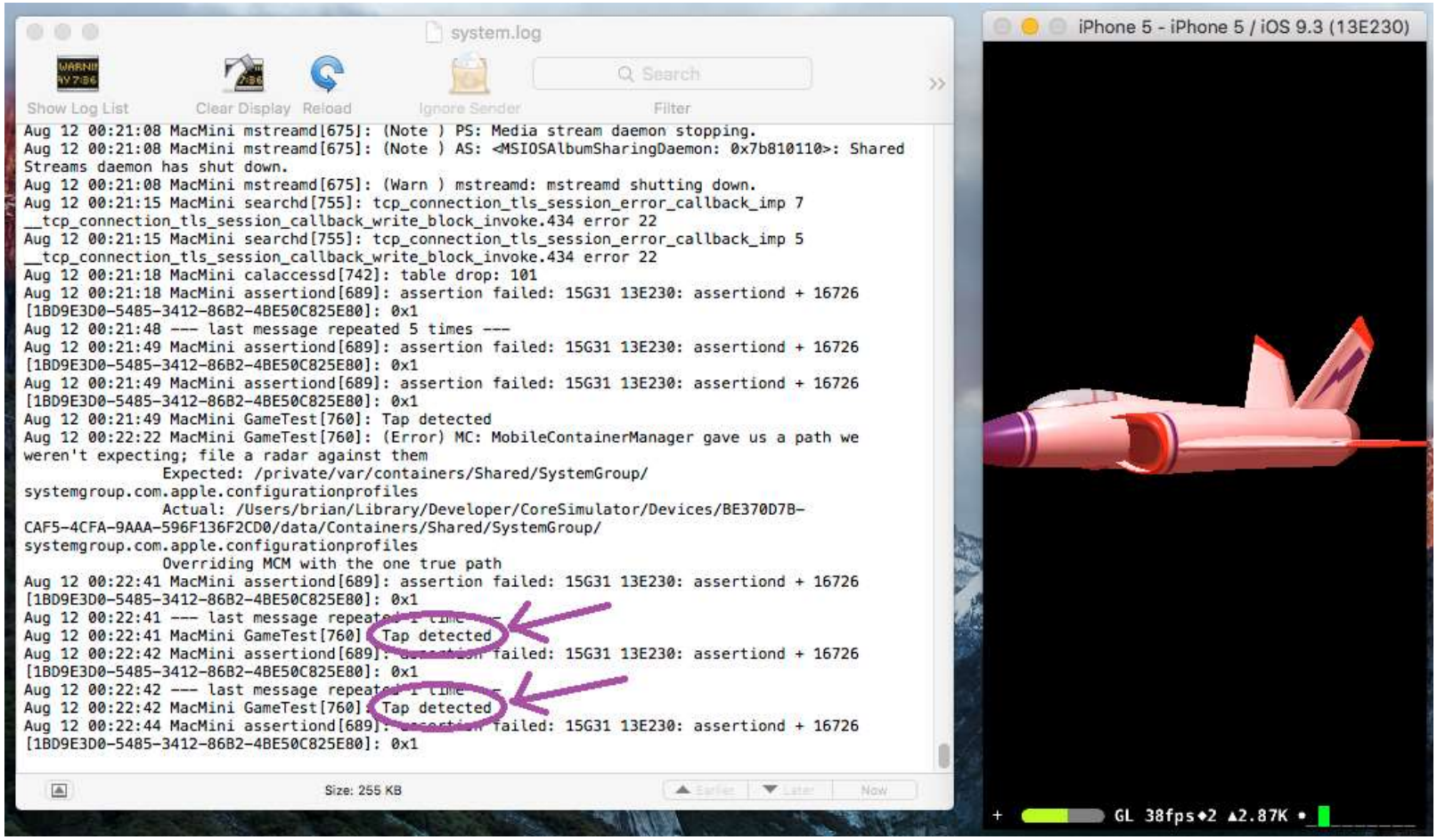
Logging

- Execution flow tracing
  - Trace messages, per OS:
    - Win[32|64] – `OutputDebugString` (Windows unit)
    - iOS – `NSLog` (iOSapi.Foundation unit)
    - macOS – `Writeln` (or `NSLog` if you ~~define~~copy it....)
    - Android – `LOG[I|W|E|F]` (Androidapi.Log unit)
  - Trace messages, cross-platform:
    - FMX – `Log.d` – 4 cross-platform overloads (FMX.Types unit):
      calls `IFMXLoggingService.Log`
      FMX – `Log.TimeStamp`, `Log.Trace`

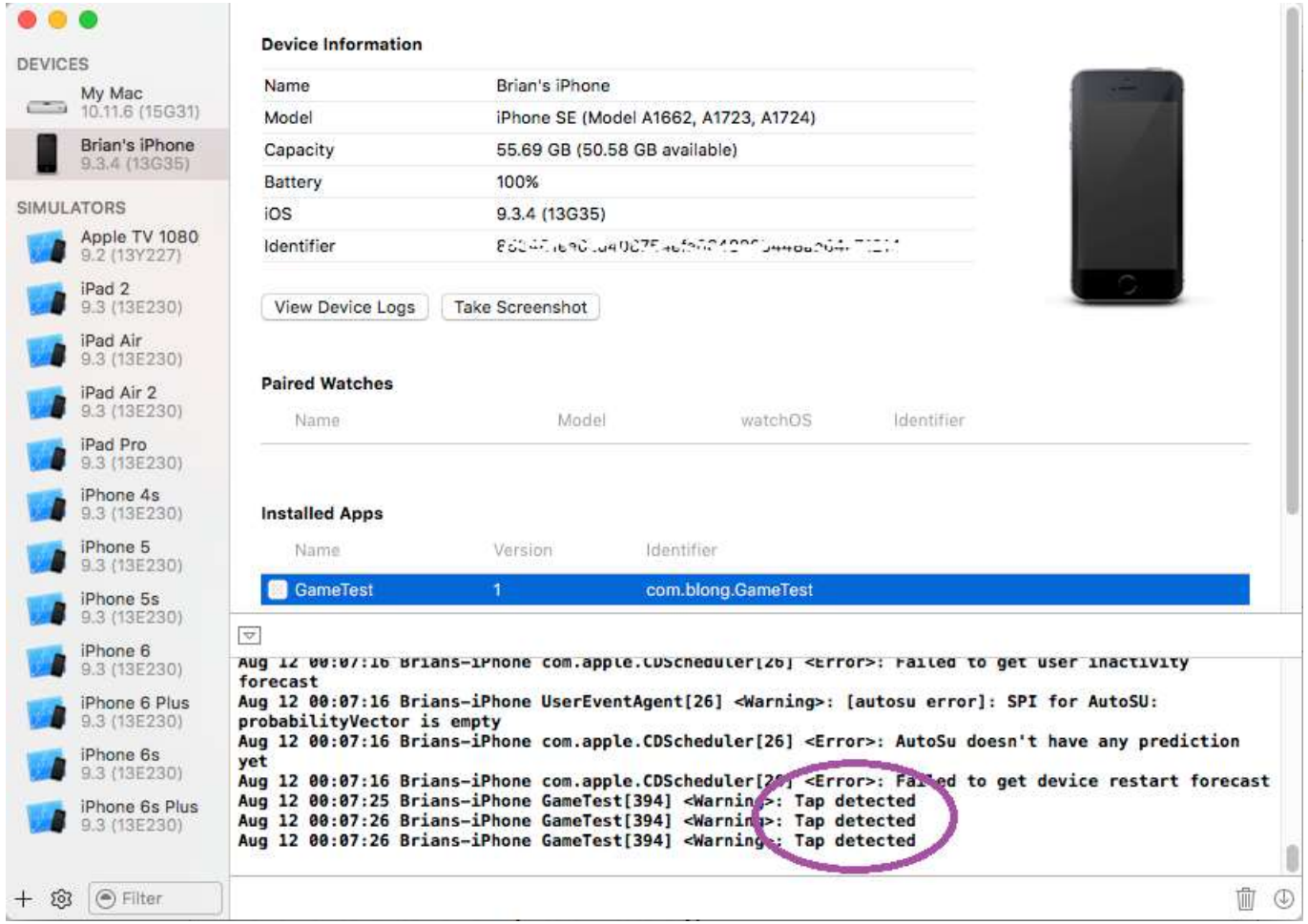
Logging

- Windows
  - Delphi's Event Log
  - Sysinternals DebugView
- Android:
  - DDMS (from Android SDK) - deprecated
  - Monitor (from Android SDK)
- iOS devices:
  - Xcode
- macOS, iOS Simulator:
  - Console(.app)

Logging


system.log
Show Log List
Clear Display
Reload
Ignore Sender
Search
Filter
Aug 12 00:21:08 MacMini mstreamd[675]: (Note) PS: Media stream daemon stopping.
Aug 12 00:21:08 MacMini mstreamd[675]: (Note) AS: <MSIOSAlbumSharingDaemon: 0x7b810110>: Shared Streams daemon has shut down.
Aug 12 00:21:08 MacMini mstreamd[675]: (Warn) mstreamd: mstreamd shutting down.
Aug 12 00:21:15 MacMini searchd[755]: tcp_connection_tls_session_error_callback_imp 7 __tcp_connection_tls_session_callback_write_block_invoke.434 error 22
Aug 12 00:21:15 MacMini searchd[755]: tcp_connection_tls_session_error_callback_imp 5 __tcp_connection_tls_session_callback_write_block_invoke.434 error 22
Aug 12 00:21:18 MacMini calaccessd[742]: table drop: 101
Aug 12 00:21:18 MacMini assertiond[689]: assertion failed: 15G31 13E230: assertiond + 16726 [1BD9E3D0-5485-3412-86B2-4BE50C825E80]: 0x1
Aug 12 00:21:48 --- last message repeated 5 times ---
Aug 12 00:21:49 MacMini assertiond[689]: assertion failed: 15G31 13E230: assertiond + 16726 [1BD9E3D0-5485-3412-86B2-4BE50C825E80]: 0x1
Aug 12 00:21:49 MacMini assertiond[689]: assertion failed: 15G31 13E230: assertiond + 16726 [1BD9E3D0-5485-3412-86B2-4BE50C825E80]: 0x1
Aug 12 00:21:49 MacMini GameTest[760]: Tap detected
Aug 12 00:22:22 MacMini GameTest[760]: (Error) MC: MobileContainerManager gave us a path we weren't expecting; file a radar against them
Expected: /private/var/containers/Shared/SystemGroup/systemgroup.com.apple.configurationprofiles
Actual: /Users/brian/Library/Developer/CoreSimulator/Devices/BE370D7B-CAF5-4CFA-9AAA-596F136F2CD0/data/Containers/Shared/SystemGroup/systemgroup.com.apple.configurationprofiles
Overriding MCM with the one true path
Aug 12 00:22:41 MacMini assertiond[689]: assertion failed: 15G31 13E230: assertiond + 16726 [1BD9E3D0-5485-3412-86B2-4BE50C825E80]: 0x1
Aug 12 00:22:41 --- last message repeated 1 time ---
Aug 12 00:22:41 MacMini GameTest[760]: Tap detected
Aug 12 00:22:42 MacMini assertiond[689]: assertion failed: 15G31 13E230: assertiond + 16726 [1BD9E3D0-5485-3412-86B2-4BE50C825E80]: 0x1
Aug 12 00:22:42 --- last message repeated 1 time ---
Aug 12 00:22:42 MacMini GameTest[760]: Tap detected
Aug 12 00:22:44 MacMini assertiond[689]: assertion failed: 15G31 13E230: assertiond + 16726 [1BD9E3D0-5485-3412-86B2-4BE50C825E80]: 0x1
Size: 255 KB
iPhone 5 - iPhone 5 / iOS 9.3 (13E230)
GL 38fps 2 ▲2.87K

Logging


DEVICES
My Mac
10.11.6 (15G31)
Brian's iPhone
9.3.4 (13G35)
SIMULATORS
Apple TV 1080
9.2 (13Y227)
iPad 2
9.3 (13E230)
iPad Air
9.3 (13E230)
iPad Air 2
9.3 (13E230)
iPad Pro
9.3 (13E230)
iPhone 4s
9.3 (13E230)
iPhone 5
9.3 (13E230)
iPhone 5s
9.3 (13E230)
iPhone 6
9.3 (13E230)
iPhone 6 Plus
9.3 (13E230)
iPhone 6s
9.3 (13E230)
iPhone 6s Plus
9.3 (13E230)
Filter
Device Information
Name Brian's iPhone
Model iPhone SE (Model A1662, A1723, A1724)
Capacity 55.69 GB (50.58 GB available)
Battery 100%
iOS 9.3.4 (13G35)
Identifier
View Device Logs
Take Screenshot
Paired Watches
Name Model watchOS Identifier
Installed Apps
Name Version Identifier
GameTest 1 com.blong.GameTest
forecast
Aug 12 00:07:16 Brians-iPhone UserEventAgent[26] <Warning>: [autosu error]: SPI for AutoSU:
probabilityVector is empty
Aug 12 00:07:16 Brians-iPhone com.apple.CDScheduler[26] <Error>: AutoSu doesn't have any prediction
yet
Aug 12 00:07:16 Brians-iPhone com.apple.CDScheduler[26] <Error>: Failed to get device restart forecast
Aug 12 00:07:25 Brians-iPhone GameTest[394] <Warning>: Tap detected
Aug 12 00:07:26 Brians-iPhone GameTest[394] <Warning>: Tap detected
Aug 12 00:07:26 Brians-iPhone GameTest[394] <Warning>: Tap detected

Logging

- Taking it further
  - Subverting the automation system:
    - `VarDispProc`
  - ~~Subverting the assertion system (debug vs. release build):~~
    - ~~`AssertErrorProc`~~
  - `WriteLn`, maybe with a TFDD
  - `TDebugUtils.DebugPrint` (XE3+):
    - Uses `WriteLn`
    - Ripe for the aforementioned TFDD

Debugging tricks/gotchas

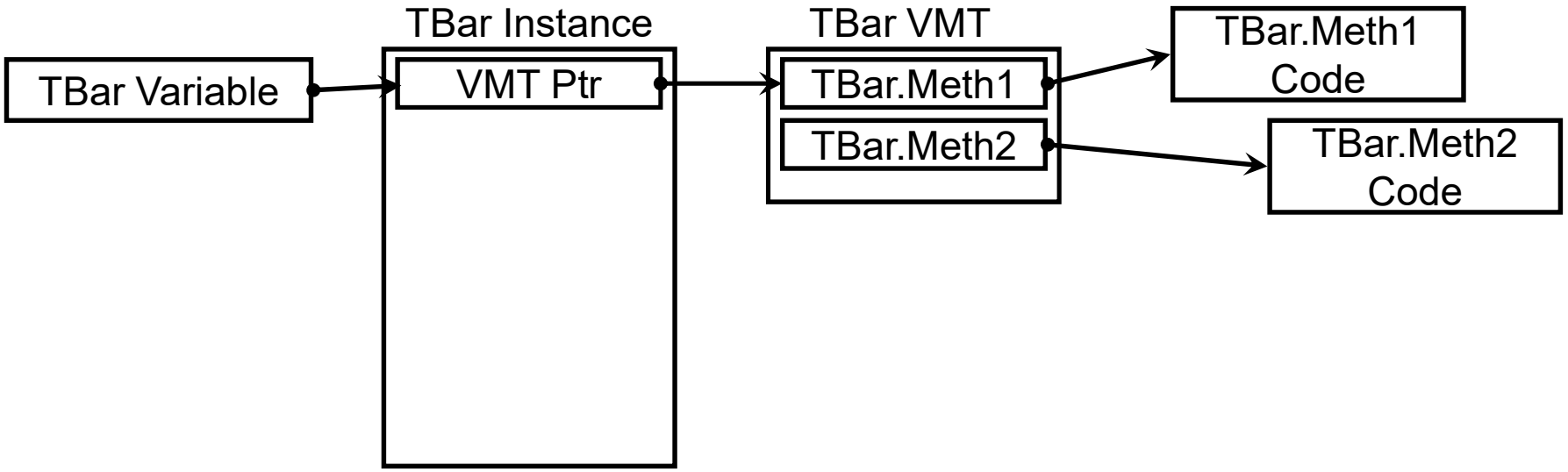
- Watches and side effects
  - side effects will silently run code
- Watch format specifiers, e.g. x, p and m
  - as per your old Turbo/Borland Pascal User's Guide
- Pass counts
  - informative as well as functional
- Breakpoint groups
  - more flexible conditional breakpoints

Debugging tricks/gotchas

- Scenario testing
  - Changing data with Evaluate/Modify
  - Changing the EIP in CPU window or dragging gutter arrow
  - NOP, DB, JMP
- Locating a dialog call
  - Pause your program (Run, program Pause)
  - Double-click the first thread in the thread list window (View, Debug Windows, Threads or `Ctrl+Alt+T`)
- Thread navigation
  - Take up the offer to name threads when creating them
  - `TThread.NameThreadForDebugging`
  - Easy to locate them in the threads list

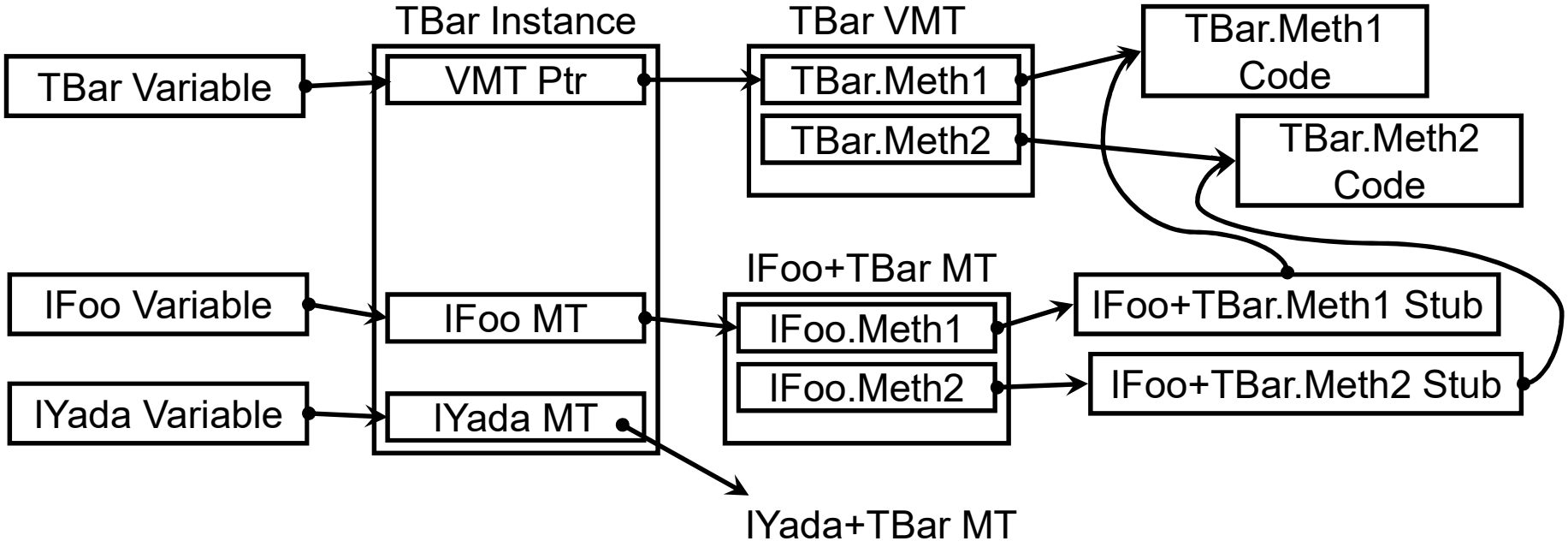
CPU window

- Code vs. data
- Object references
- Following links



CPU window

- Code vs. data
- Object references
- Following links



Chasing down AVs

- Dedicated tools
  - madExcept
  - EurekaLog
  - JEDI JCL Debug Framework
- Manually (more often than never)
  - Relies on archiving distributed builds, or using VCS
  - IDE searching
  - MAP files (Segment, Publics by Value, Line numbers)

Alternative memory manager

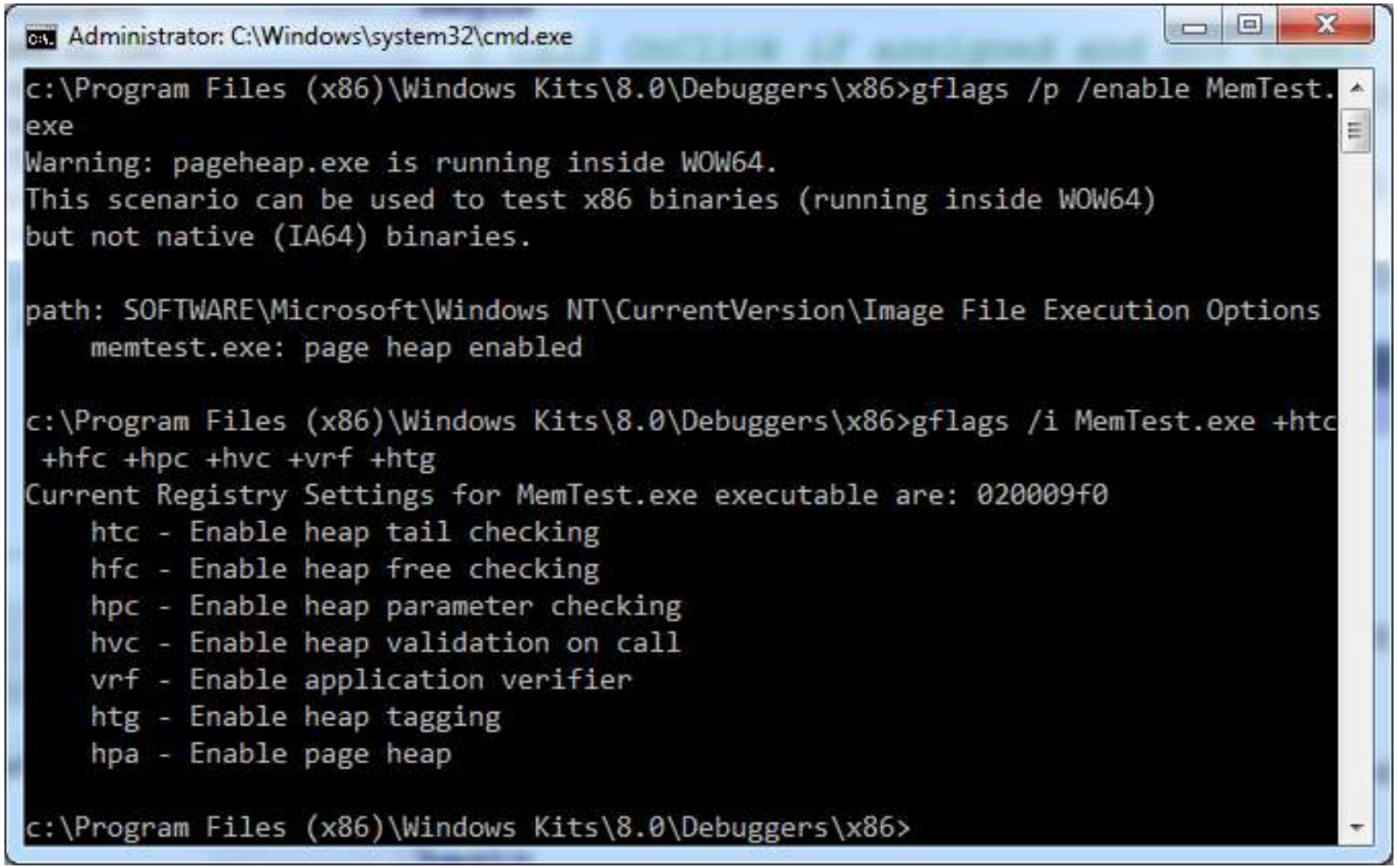
- `HeapAlloc`-based memory manager:
  - Ray Lischners's from Code central (with suitable updating): http://cc.embarcadero.com/Item/22668
  - RTL rebuild with `SIMPLEHEAP` defined (XE3+)
- Use in conjunction with GFlags settings:
  - `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\AppName.exe` aka IFEO
  - Get from Microsoft's Debugging Tools for Windows: https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger
  - Or plug in values by hand

Rebuilding the RTL*

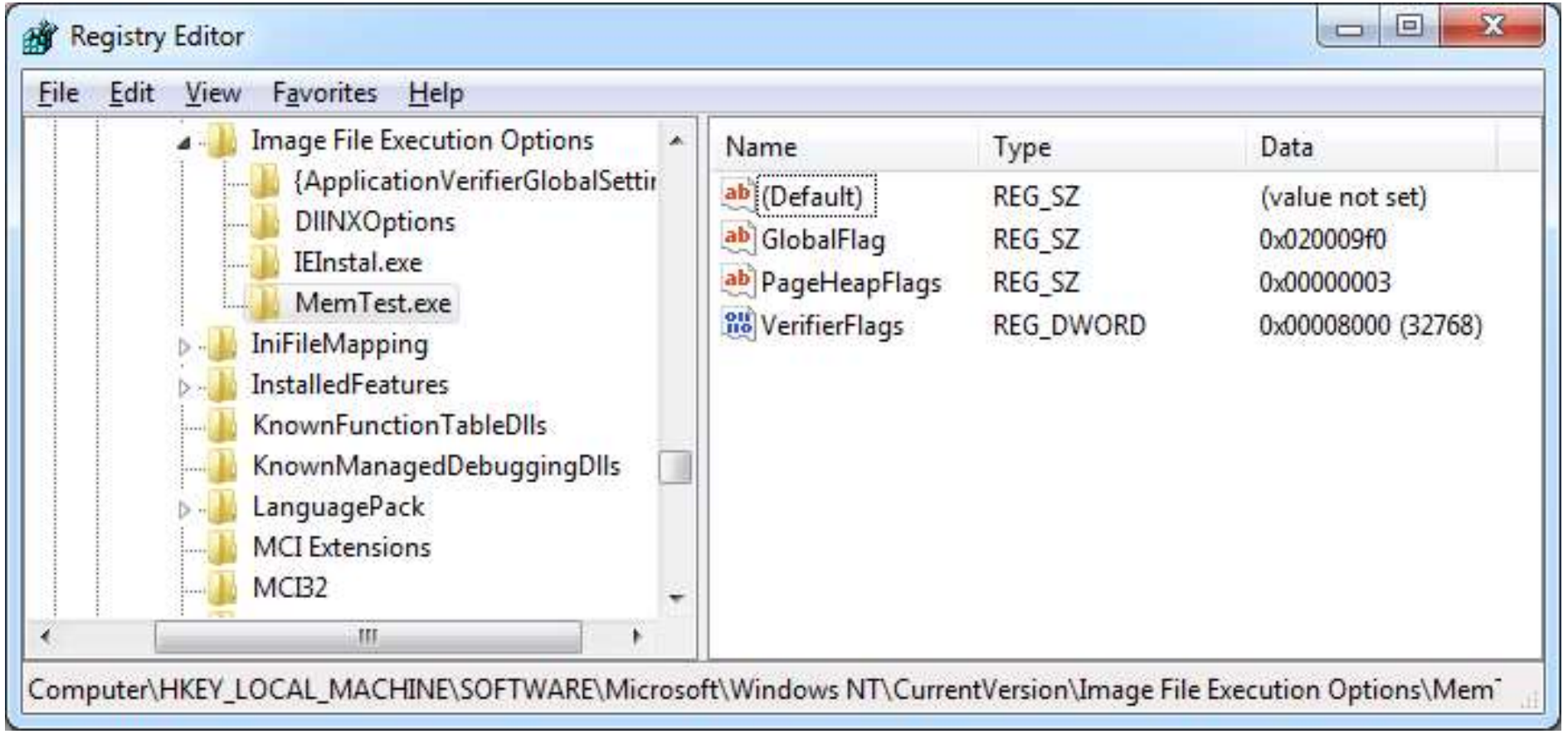
- Copy $(BDS)\source\rtl outside of Program Files (x86)
- Run:
  - `cd <RTL_folder>`
  - `set PROPERTIES=/p:DCC_Define=SIMPLEHEAP`
  - `buildrtl.bat debug`
- Output goes to:
  - %PUBLIC%\Documents\Embarcadero\Studio\19.0\lib\Win32
  - $(BDSCOMMONDIR)\lib\Win32
- Add to Debug DCU path for Win32 in BDS options

* Actually doesn't build in Berlin due to missing System.Internal.JSONHlpr unit

GFlags


Administrator: C:\Windows\system32\cmd.exe
c:\Program Files (x86)\Windows Kits\8.0\Debuggers\x86>gflags /p /enable MemTest.exe
Warning: pageheap.exe is running inside WOW64.
This scenario can be used to test x86 binaries (running inside WOW64)
but not native (IA64) binaries.
path: SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
memtest.exe: page heap enabled
c:\Program Files (x86)\Windows Kits\8.0\Debuggers\x86>gflags /i MemTest.exe +htc +hfc +hpc +hvc +vrf +htg
Current Registry Settings for MemTest.exe executable are: 020009f0
htc - Enable heap tail checking
hfc - Enable heap free checking
hpc - Enable heap parameter checking
hvc - Enable heap validation on call
vrf - Enable application verifier
htg - Enable heap tagging
hpa - Enable page heap
c:\Program Files (x86)\Windows Kits\8.0\Debuggers\x86>

GFlags


Registry Editor
File Edit View Favorites Help
Image File Execution Options
{ApplicationVerifierGlobalSettir
DllNXOptions
IEInstal.exe
MemTest.exe
IniFileMapping
InstalledFeatures
KnownFunctionTableDlls
KnownManagedDebuggingDlls
LanguagePack
MCI Extensions
MCI32
Name
Type
Data
(Default)
REG_SZ
(value not set)
GlobalFlag
REG_SZ
0x020009f0
PageHeapFlags
REG_SZ
0x00000003
VerifierFlags
REG_DWORD
0x00008000 (32768)
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Mem

FastMM4

- http://fastmm.sourceforge.net
- Like the default memory manager but better!
- Great VFM
- Checks for objects used after freed
  - `FullDebugMode`
- Checks for interface references used after the fact
  - `CatchUseOfFreedInterfaces`
- Checks for heap corruption
  - `CheckHeapForCorruption`
- Checks for memory leaks
  - `EnableMemoryLeakReporting`

Questions/Consultancy?

- brian@blong.com
- http://blong.com
- http://blog.blong.com