



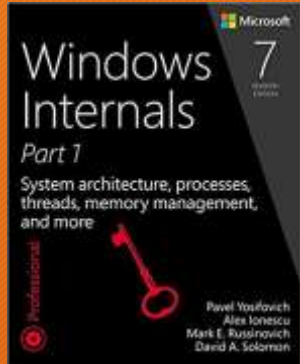
Writing Your Own Diagnostic and Profiling Tools with Event Tracing for Windows (ETW)

Pavel Yosifovich

@zodiacon

zodiacon@live.com

Something About Me



- Developer, Trainer, Author and (apparently) Speaker
- Book author
 - “Windows Internals 7th edition, Part 1” (co-author, 2017)
 - “WPF 4.5 Cookbook” (2012)
 - “Windows Kernel Programming” (WIP, 2019)
- Pluralsight author
- Author of several open-source tools (<http://github.com/zodiacon>)
- Blogs: <http://blogs.microsoft.co.il/pavely>, <http://scorpiosoftware.net>

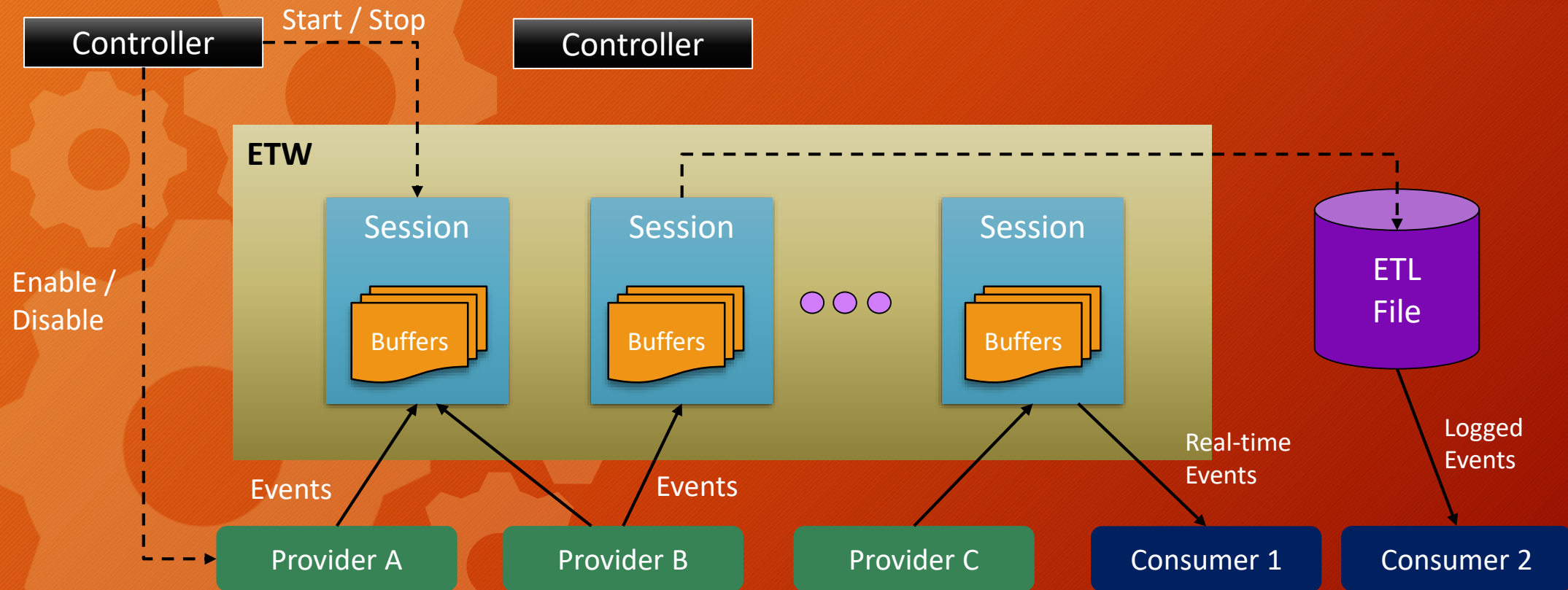
Agenda

- Introduction to ETW
- Tools
- Libraries
- Demos
- Summary
- Q & A

Event Tracing for Windows (ETW)

- Introduced in Windows 2000
- Event Tracing / Logging mechanism
 - Low overhead even with high event volume
- Traces can be recorded to a file and/or to a real-time session
- System-wide
- Lots of registered providers out of the box
 - `c:> logman query providers`

ETW Architecture





ETW Sessions

Demo



Tools for Working with ETW

- Xperf, Tracerpt, Logman
 - Built-in tools for recording ETW / reporting / formatting
- Windows Performance Recorder (WPR) / Analyzer (WPA)
 - Part of the Windows Performance Toolkit (Windows SDK)
- TraceView
 - Windows SDK and WDK
- PerfView
 - Uses ETW to analyze (mostly) .NET related issues
- ProcMonX
 - Possible replacement for ProcMon
- Your own tools!



ETW Explorer

Demo

Working with ETW in .NET

- Providers
 - `System.Diagnostics.Tracing` namespace
 - Not the focus of this talk
- Consumers
 - Open source libraries (available through Nuget)
 - `Microsoft.Diagnostics.Tracing.TraceEvent`
 - Used by PerfView
 - `Microsoft.0365.Security.Native.ETW`
 - krabs wrapper (a.k.a. “Lobsters”)



Simple Provider and Consumer

Demo

The Kernel Provider

- Requires running elevated to use
- Prior to Windows 8 – only one such session can exist
 - Name of session must be “NT Kernel Logger”
 - `KernelTraceEventParser.KernelSessionName` has this name
- Many interesting events
 - Mostly “documented” in MSDN



Consuming Kernel Provider Events

Demo

TraceEvent and Parsers

- The `Diagnostics.Tracing.TraceEvent` library comes with several built-in parsers
 - E.g. Kernel and CLR
- For other providers, a tool can generate the required parser code based on the XML manifest
 - `TraceParserGen.exe`



Parser Generator

Demo



The CLR Provider

- Events published by the CLR itself
- Use `ClrTraceEventsParser` to consume
- Register for interesting events
- Filter by the process(es) of interest
- For .NET Core (2.2 and later)
 - Use the `EventListener` class



Consuming CLR Provider Events

Demo

```
try {  
    // many lines of code  
}  
Catch {  
    // maybe log  
}
```


Call Stacks

- ETW Events can provide call stack
- The `Microsoft.Diagnostics.Tracing.Etlx.TraceLog` class does all the heavy lifting
- Must enable some kernel provider events



Call Stacks

Demo

Summary

- ETW is a low-overhead logging infrastructure
 - Common across kernel mode and user mode
- Many ETW providers out of the box
 - Research and experiment
- Easy to consume to a file or a real-time session
 - Sometimes state management is required
- Call stack is also available

Thank you!

