

# Programming in C# Jump Start

Jerry Nixon | Microsoft Developer Evangelist

Daren May | President & Co-founder, Crank211

# 06 | Advanced C#, Part 2

Jerry Nixon | Microsoft Developer Evangelist

Daren May | President & Co-founder, Crank211

# Module Overview

- Splitting Assemblies and Projects
- Diagnostics and Instrumentation

# Splitting Assemblies and WinMD

MVA

Microsoft  
Virtual  
Academy

# What is an assembly?

- An assembly is a container for a set of resources and types.
- Assemblies can be referenced by other assemblies.
- Some assemblies are specific to certain technologies.
- In Visual Studio, an assembly equates to a Project

# Assembly versus Namespace

## Assembly

- An assembly is a grouping or physical container of code
- Can contain one or more namespace
- Can be granted security permissions
- Can be digitally signed
- Defines a scope

## Namespace

- A namespace is a grouping or logical container of code
- Can span multiple files
- Can span multiple assemblies
- Defines a scope

# Why share an assembly?

- Reuse. Shared assemblies allow developers to encapsulate their code so they can be reused across applications and systems.
  - Building an API (Product)
  - Line of Business applications
  - Different Platforms
    - Windows Desktop Applications
    - Windows Store Apps
    - Windows Phone Apps
    - Windows Embedded
    - Microsoft Silverlight
    - Microsoft XBox

# How to share assemblies?

- Class Libraries are the Visual Studio project type that creates an assembly that can be shared
- Portable Class Libraries are the Visual Studio project type that creates an assembly that can be shared across platforms
  - uses only the lowest-common-denominator.
- Windows Component Libraries (WinMD/Windows Metadata) are the Visual Studio project type that creates an assembly for Windows Store Apps that can be shared across languages.
  - JavaScript cannot be used to create WinMDs



# What is a Windows Runtime Component? (035)

- A Windows Runtime Component is a code library can be consumed by all App Store development languages
  - Through projection

## Simple Rules

- Only WinRT types can be exposed
- No generics can be exposed
- Name can't start with Windows
- Polymorphism isn't available
- Public classes must be sealed
- Public structures can only have
  - Public members, properties, methods, events

DEMO



# Windows Runtime Components

# Diagnostics and Instrumentation

MVA

Microsoft  
Virtual  
Academy

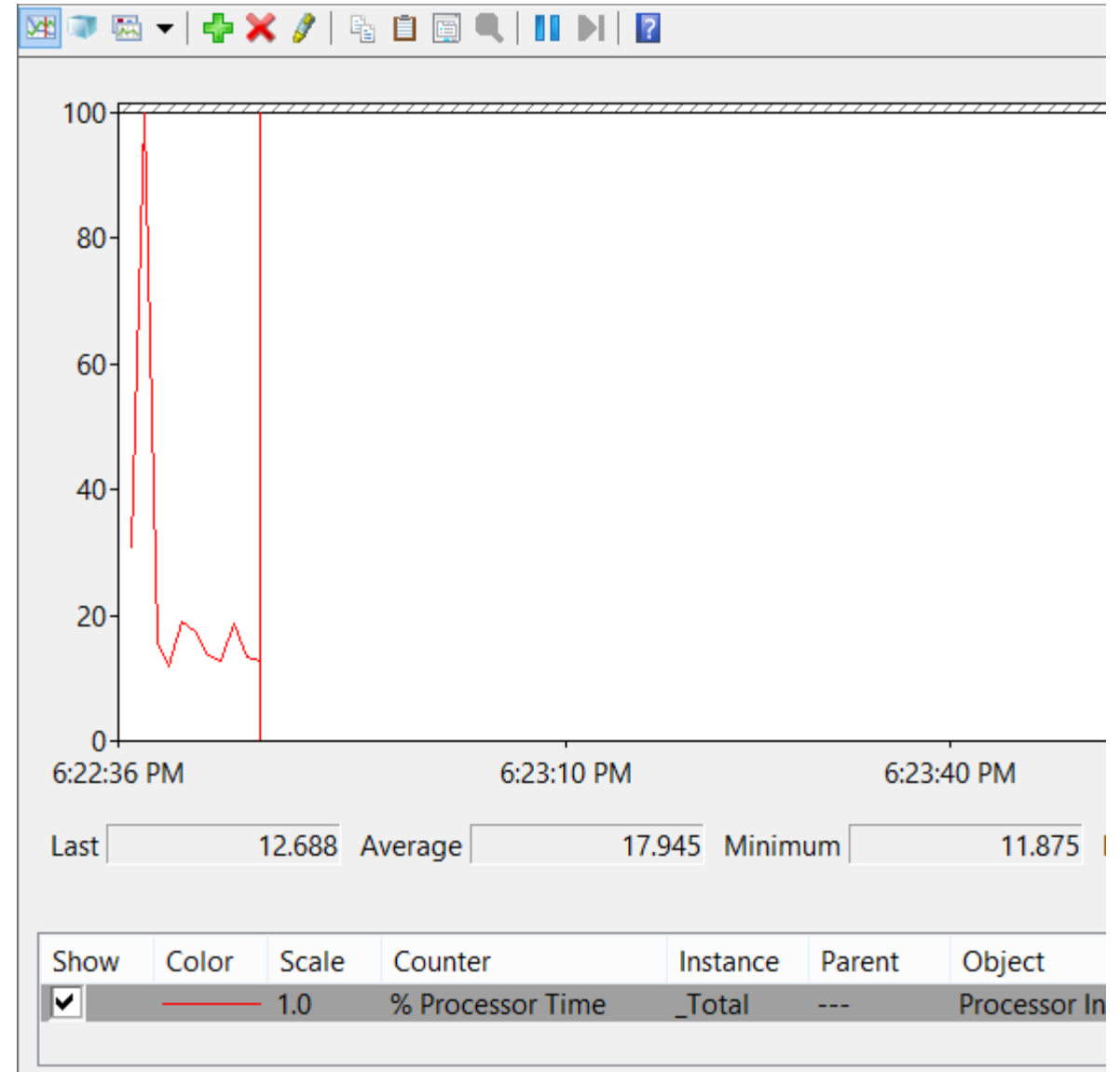
# What is Instrumentation?

- *Instrumentation* is code that reports performance information.
- *Telemetry* aggregates *instrumentation* for *analysis*.
- *Diagnostics* or *Analysis* is the use a *telemetry* to track causes of errors or identify trends.

# What is a Performance Counter?

- Performance Counter is a sampling for a single operation
- You write to a counter by incrementing or decrementing
  - Trending is accomplished using tooling
- You must have permission
- Counters are typically categorized

```
var perfCounter = new PerformanceCounter
{
    CategoryName = category,
    CounterName = counter,
    MachineName = machine,
    ReadOnly = false,
};
perfCounter.IncrementBy(10);
```



# Other diagnostic techniques

- **Tracing – informative messages during execution**
  - Strategic placement
  - Fail/Assert – adds text to output (condition); implies error
  - Write/Writelf – adds text to output (condition)
  - WriteLine/WriteLinelf – adds text to output w/newline (condition)
- **Volume control (Trace Listeners & Trace Switch)**
- **Event logs – receive and track major events during execution**
  - Windows
  - Custom

```
public void Run()
{
    Write("Demo", "Hello World",
        System.Diagnostics.EventLogEntryType.Information);
}

void Write(string app, string message,
    System.Diagnostics.EventLogEntryType type)
{
    System.Diagnostics.EventLog
        .WriteEntry(app, message, type);
}
```

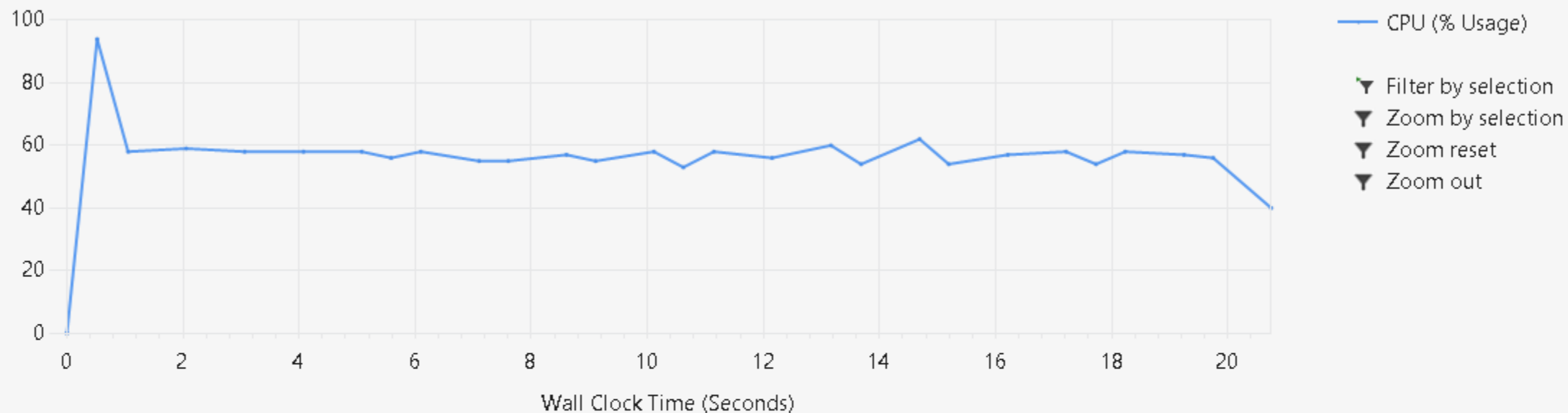


# What is Performance Profiler? (026)

- Performance Profiler is a tool built into Visual Studio
  - conducts *Performance Sessions*
- Performance Session is an execution of your application during which Performance Counters are collected.
  - Sampling method
  - Sampling interval
- A Performance Report allows developers to identify high-cost operations.
  - Call Frequency
  - Call Duration

## Instrumentation Profiling Report

19.4 seconds of total execution time



## Hot Path

| Function Name                                | Elapsed Inclusive Time % | Elapsed Exclusive Time % |
|--|--------------------------|--------------------------|
| <code>After026.Program.Main(string[])</code> | 100.00                   | 0.02                     |
| <code>After026.Program.Run3()</code>         | 35.81                    | 0.00                     |
| <code>After026.Program.Run5()</code>         | 27.36                    | 0.00                     |
| <code>After026.Program.Run2()</code>         | 20.25                    | 0.00                     |
| <code>After026.Program.Run4()</code>         | 14.02                    | 0.00                     |

DEMO



Diagnostics

# Module Recap

- Splitting Assemblies and Projects
- Diagnostics and Instrumentation



©2013 Microsoft Corporation. All rights reserved. Microsoft, Windows, Office, Azure, System Center, Dynamics and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.