

Coding for Debugging

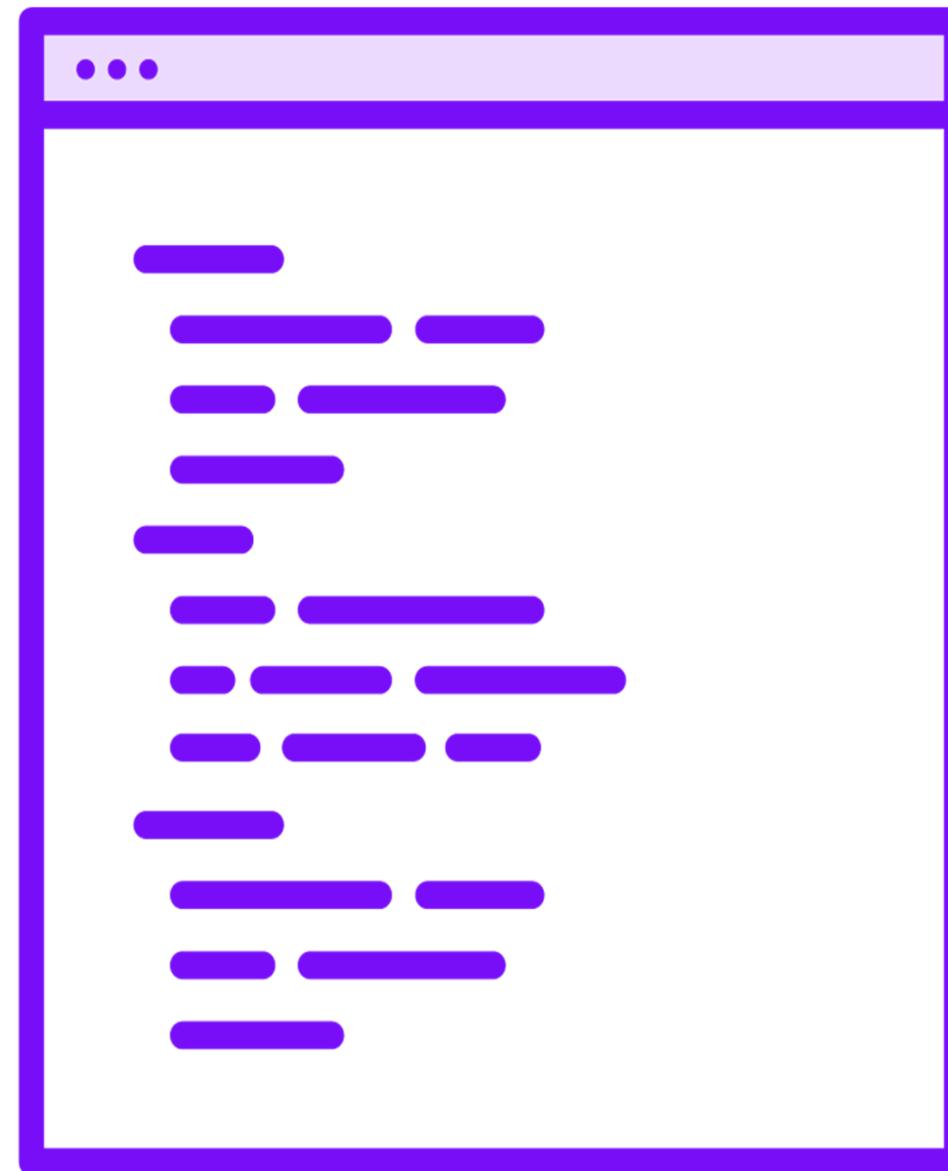


Neil Morrissey

Solutions Architect

linkedin.com/in/neilmorrissey | www.neilmorrissey.net

System.Diagnostics.Debugger



Debugger.Launch()

- Choose a debugger
- Code initiates Debugging

Debugger.Break()

- Breaks at the line of code

Debugger.IsAttached



Module Overview



System.Diagnostics.Debugger

Debug.Assert

Attributes

- DebuggerHidden
- DebuggerDisplay

Logging to assist debugging





Debug.Assert(<condition>)

Typically used to verify input parameters

Useful in team development

When condition fails – break at Assert

Can include message to TraceListeners

Debug.Assert removed from release compilation

Provide documentation on intention of code

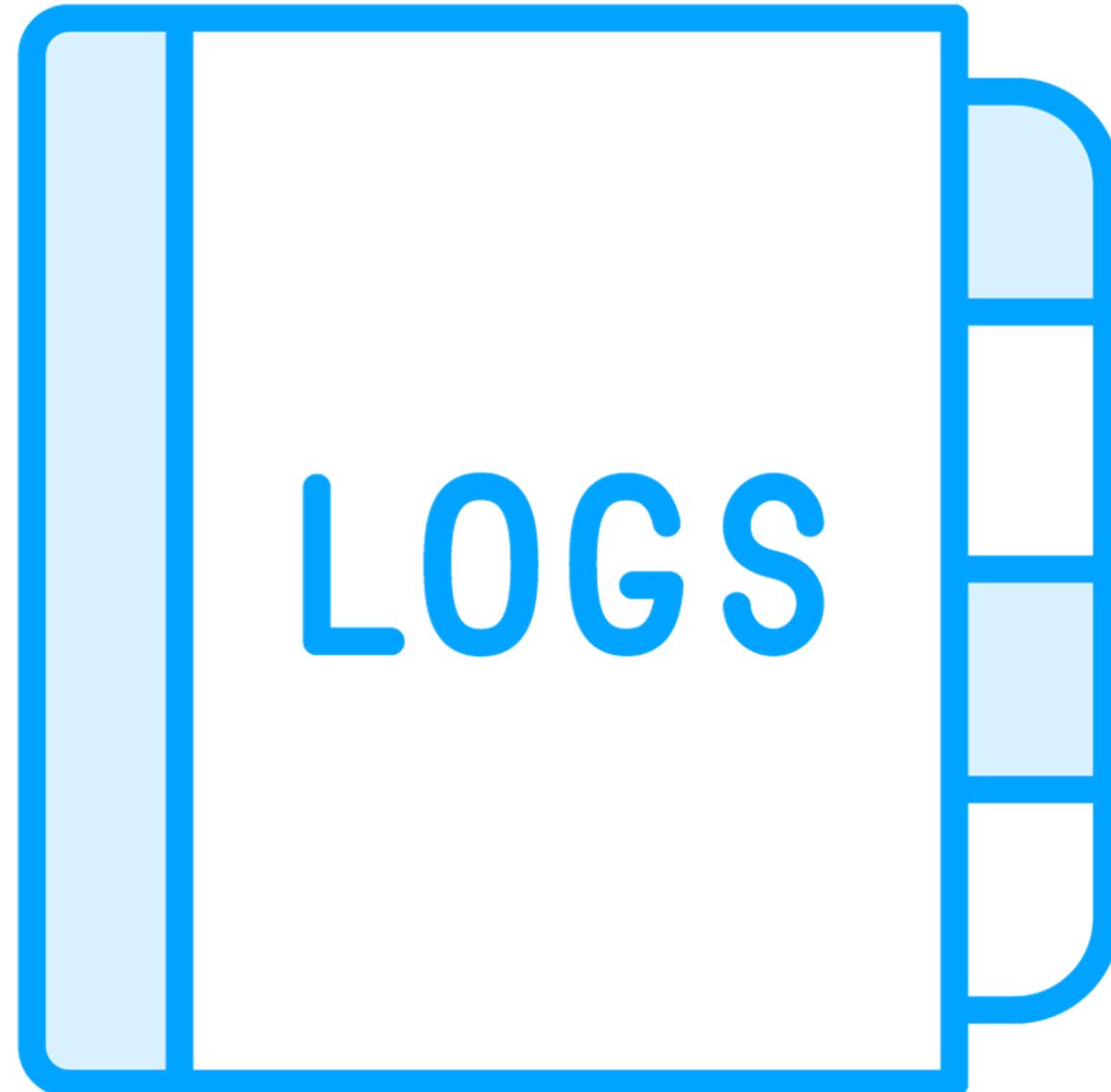


DebuggerHidden Attribute



Logging in .NET Applications





Debug.WriteLine

- Removed from release version of app

Trace.WriteLine

- Stays in release compilation

iLogger

- Abstracts logging implementation
- Log destinations are Logging Providers



Built-in Logging Providers

Console

Debug

EventLog

EventSource



Additional Logging Providers

Azure App Service
(file system)

Azure App Service
(blob storage)

Application Insights

3rd Party Logging
Providers

Custom Logging
Provider





ILogger demo

- Log to Console
- Log to Windows Event Log

Create EventLog Event Source

Configure EventLog Logging Provider

Log Traces, Information, Errors in code

ILogger configuration in appsettings.json

View logging output



Course Summary



Debugger basics

Debugging an exception and a defect

Benefits of becoming a better debugger

Approach to debugging

More debugging features of VS, VS Code

Advanced breakpoints

System.Diagnostics

Logging to help with debugging



Thank You!



Neil Morrissey

www.neilmorrissey.net

www.linkedin.com/in/neilmorrissey/

