

.NET Debugging Workshop

DNDebug - Version: 3.3

 **3 days Course**

Description:

The famous quote goes: "Debugging code is twice as hard as writing it in the first place." Can we prove otherwise? Debugging has become more complicated as new technologies emerged, but the .NET framework and the CLR offer us new debugging capabilities that we have never had before. External tools and extensions augment the built-in debugging capabilities and take us to completely new levels of understanding application behavior, even in a production environment. Realizing the full potential of all these possibilities isn't easy, but practice makes perfect and the exercises sprinkled throughout this three-day course will give participants valuable take-aways for their day jobs.

Intended audience:

Experienced .NET programmers and integrators who seek to extend their debugging toolbox with new techniques and practice them first-hand.

Prerequisites:

- Working knowledge of C# 3.0
- Working knowledge of the .NET Framework
- Basic understanding of the inner workings of the .NET Framework
- Familiarity with computer organization concepts: CPU cache, registers, main memory

Objectives:

- Leveraging advanced Visual Studio debugging techniques
- Generating and analyzing application crash and hang dumps
- Solving memory leaks, deadlocks, and crashes using WinDbg and SOS
- Using external debugging and monitoring tools

Topics:

Exceptions and Dumps

- Exception Handling
- Debugging Symbols
- Dump Files and Types
- Generating Dumps
- Automatic Dump Generation
- Opening Dump Files

Introduction to WinDbg

- Basic WinDbg Commands
- Smart Breakpoints
- WinDbg Scripts
- WinDbg Extensions
- LAB: Getting Acquainted with WinDbg
- LAB: Capturing Crash Dumps (x3)

Debugging Tools

- Performance Counters
- Process Explorer
- Process Monitor
- Application Compatibility Toolkit
- ETW and Xperf
- GFlags
- LAB: Profiling with Xperf
- LAB: Process Monitor
- LAB: Application Compatibility Toolkit

Debugging in Visual Studio

- Visual Studio Windows
- Breakpoints and Tracepoints
- Data Breakpoints, Function Breakpoints
- Threads
- Parallel Stacks, Parallel Tasks
- Static Code Analysis
- LAB: Runtime Checks

SOS

- Setting Smart Breakpoints
- Analyzing Memory Leaks
- Inspecting Objects
- Inspecting Threads and Stacks
- Advanced Commands
- LAB: Getting Acquainted with SOS
- LAB: Capturing Crash Dumps (x2)
- LAB: Deadlock (x2)
- LAB: Memory Leak (x4)

.NET Debugging Tools

- Managed Debugging Assistants
- IntelliTrace
- Visual Studio Profiler
- CLR Profiler
- ANTS Memory Profiler
- Assembly Loading Diagnostics
- LAB: Fusion Diagnostics
- LAB: IntelliTrace

° Assembly Language Fundamentals

° Interop Debugging

CLR Internals

- Reference Types and Value Types
- Boxing and Unboxing
- Implementing Value Types Correctly
- GC Heap and Flavors
- Generational GC
- GC Segments
- Finalization