Windows 7 Debugging & Dump Analysis

Introduction

This 5-day course gives developers and support engineers the knowledge to effectively troubleshoot Windows 7-based solutions, using a variety of system-level tools. It presents knowledge to locate and isolate Win7 kernel bugs as well as user-mode application bugs. Both "live" and post-mortem techniques for troubleshooting are presented. The course is designed for Windows 7 operating systems and based on the RTM version of Windows 7.

Course Objectives

At course completion, students should have the following knowledge and skills:

- Understand the Windows 7 family of debuggers and tools
- Understand the role of the Portable Executable file format for executables
- Use symbol files in PDB or DBG format
- Use and understand DEBUG and RELEASE builds
- Use link map files
- Using the Visual Studio debugger, perform source level tracing
- Set up advanced breakpoints
- Use the full set of Visual Studio debugging facilities
- Manage multiple threads during debugging
- Use the debug output window
- Use the NTSD debugger for source-level debugging
- Perform remote debugging using Remote and CDB
- Understand the structure of the Intel & Windows stack
- Perform stack traces
- Recovering information from a torn stack
- Use WinDbg to perform source level debugging of a kernel-mode device driver
- Perform a kernel mode crash dump analysis
- Collect and analyze data from a user mode crash using Dr. Watson, Adplus, and WinDbg

Prerequisites

Before taking this course, students should have the following skills:

- Operating system concepts such as
  - Memory management
  - Resource management
  - Reentrancy
  - File system management
- Debugging concepts and some techniques
- Preferable: Experience with Windows programming (C, C++, etc.)

Course Structure

This course is a lectured seminar with some hands-on debugging exercises. Lectures include numerous demonstrations.

Course Outline

Windows 7 Architecture

- History of Windows OS
- Design Goals
- Features of the OS
- Threads
- Processes
- Client/Server Architecture

Debuggers & Environment

- The Windows Debuggers
- The Portable Executable (PE) File Format
- Symbol Files
- Map Files
- Debug & Release Builds

Visual Studio Debugging

- Source File Debugging
- Setting Breakpoints
- The Debug Windows
- Thread Management
- Exception Management
- Remote Debugging

Memory Management

- Virtual Address Translation
- Page Faults
- Working Set Management
- Physical Memory Management

DLL Architecture & Debugging

- DLL Architecture
- DLL Linkage
- Imports & Exports
- Utilities for DLL management
- DLL Load Order
- Binding & Basing
- DllMain

NTSD

- NT Symbolic Debugger Features
- NTSD Command Line
- Working with Symbols
- Debugging Multiple Processors
- Using NTSD with Remote

Stack Debugging

- Structure of the Intel Stack
- Stack Optimizations
- Stack Traces
- Stack Corruption
- Stack Recovery

WinDbg

- Features of WinDbg
- WinDbg Interface
- Debug windows
- Symbol file specification
- Source file specification
- Setting breakpoints
- Controlling code execution

Windows 7 Driver Architecture

- The Windows 7 I/O Model
- I/O Processing
- The Cache Manager
- Types of Supported Device Drivers
- Driver Operation
- Plug-and-Play Manager
- Power Manager

Kernel-mode Debugging

- Overview of kernel debuggers
- Kernel mode debugging environment
- Host configuration
- Target configuration
- Symbol files
- Using WinDbg on the Host

Dump File Analysis

- Why Windows crashes

- Memory Dump Options
- Analyzing a Crash Dump with WinDbg
- User mode dump files
- An Overview of Dr. Watson
- Building an application for use with Dr. Watson
- Using ADPlus
- Capturing and analyzing a user mode crash

Hardware Debugging

- Probing the Hardware
- Accessing IO ports
- Reading & Writing Device Memory
- Viewing the Busses
- Examining Device Memory
- MP Information
- Interrupt Information
- Power Management Information

Extended Crash Dump Analysis

- Kinds of Dump Files
- Kinds of Crashes
- What Can & Can't Be Learned
- Using DumpChk
- When You Have & Don't Have Source
- Checked Builds
- Online Crash Analysis

Symbol Server

- The Problem of Symbol Files
- Symbol Server
- Symbol Store
- How Symbols are Located
- Multiple Symbol Servers
- Symbol Storage Organization
- SymStore Command Line Syntax

Driver Stress Testing

- Driver Verifier
- Buffer Boundary Conditions
- DIO Problems
- Multithread Usage Problems
- Canceled IRP Problems
- Timing Windows