

INTRODUCTION

Windows Driver Foundation (WDF) is the Microsoft unified driver model. It supports the creation of object-oriented kernel-mode and user-mode drivers for Windows. By using WDF, driver engineers can focus on their device hardware, instead of on the operating system. WDF simplifies driver development and maintenance in a number of ways, including:

Managing most interactions with the operating system, Enabling a broader range of devices to be supported by user-mode drivers.

Supporting a robust, well-designed object model, Providing intelligent default handling for common features such as Plug and Play and power management, Reducing the occurrence of common race conditions.

WDF consists of two components, the kernel-mode driver framework (KMDF) and the user-mode driver framework (UMDF), which are used to implement kernel-mode or user-mode drivers, respectively. It also includes several related testing and debugging tools.

Day 1 – Introduction to Windows Driver Development and the WDF

- Introduction to Windows Driver Development
- Brief Overview of WDM
- WDM Concepts for WDF
 - Driver Types
 - Device and Driver Stacks
 - I/O Request Packets (IRPs)
- Quick Introduction to WDF
 - Kernel Mode Driver Framework (KMDF)
 - User Mode Driver Framework (UMDF)
 - Differences between KMDF and UMDF
 - When to use KMDF and UMDF?
 - Examples of devices suitable for KMDF and UMDF
- Requirements of A Simple WDF Driver
- Code walkthrough of WDM and WDF drivers for comparison

Day 2 – The Kernel Mode Driver Framework

- Interactions between the KMDF and drivers
- Introducing the Basic KMDF Objects
- Introducing the Special KMDF Objects
- Framework Object Lifecycle
- The Object Context Space
- Handling I/O Requests
- Processing an I/O Request
- Handling Operations on File Objects
- Using I/O Targets
- Using Interrupts, Timers and Memory Buffers

Day 3 – The User Mode Driver Framework

- Introducing the UMDF and benefits of UMDF drivers
- Supported devices and device classes
- UMDF Architecture and the use of Component Object Model (COM)*

- Key Elements – Driver host process, Reflector, Driver Manager
- UMDF Objects and Object Hierarchy
- Managing UMDF Object Lifecycle
- Initializing UMDF drivers
- Callback Objects
- Processing an I/O request
- Using I/O Targets

Day 4 – Advanced Issues in WDF Drivers**

- Supporting PnP and Power Management
- Synchronization Techniques
- Handling DMA
- Supporting WMI

Day 5 – Debugging and Installing WDF Drivers

- Debugging KMDF Drivers – Problems and techniques
- KMDF Debugging Scenarios
- Installing KMDF Drivers
- Debugging UMDF Drivers
- User-Mode Debuggers
- UMDF Debugging Scenarios
- Installing UMDF Drivers

Notes

** - Participants will need to be familiar with the Component Object Model (COM) to the extent of being able to write simple in-process COM server and client applications.*

*** - These topics are very heavy and participants must be capable of sustaining the information overload if these are to be covered in detail!*

== 0 ==