

# Windows용 WDF 드라이버 작성 -- 엑셀레이티드 에디션

Windows 드라이버 파운데이션(WDF)은 Windows 드라이버 작성에 대한 최신의 표준으로, Windows용 새 드라이버를 구현하는 데에 가장 선호되는 방법이다.

신속하고 집중적인 이 세미나는 WDF 커널 모드 드라이버 프레임워크 (KMDF)를 사용하여, Windows 드라이버를 설계, 개발 또는 시험하는 방법의 이해가 필요한 엔지니어용으로 마련되었다. 세미나는 KMDF 소프트웨어 드라이버, 필터 드라이버, 범용 직렬 버스(USB) 장치 드라이버, 프로그램식 I/O 형 주변장치 연결 인터페이스(PCI) 드라이버에 특히 초점을 맞추고 있다. 하지만, WDF 기본 개념을 이해하고 나면, 사용자 모드 드라이버 프레임워크(UMDF)를 사용하여 다른 클래스의 장치용 KMDF 드라이버 작성은 상대적으로 간단히 만들 수 있게 된다.

이 특별 엑셀레이티드 세미나에서는 다른 OSR 세미나보다 더욱 심화된 내용의 KMDF를 다룬다. KMDF 확장의 논의를 통하여, 추가적인 모듈 전체를 특정 네트워크(NDIS)와 오디오/비디오(커널 스트리밍) 장치에도 추가하는 방법도 논의된다.

이 세미나는 Windows 드라이버 모델 (WDM)을 사용하여 작성하는 방법을 알고 있거나 WDF에 이미 경험이 있는 개발자는 물론, Windows 드라이버 작성 경험이 전혀 없는 개발자 모두를 위해 마련되었다.

세미나가 완료 후의 계속 교육: 교육받은 내용을 즉시 실제 적용하려는 수강생들에게는 샘플 드라이버, 문제 및 솔루션 등이 과제물로서 제시된다. USB와 PCI 실습 과제는 OSR USB FX2와 Sealevel Systems 8018 Digital I/O 키트를 통하여 즉시 사용할 수 있으며, 적당한 가격으로 [OSR 온라인 스토어](#)에서 구입될 수 있다.

## 세부 정보

기간: 4 일

형식: 강의

비용 \$2,450.00

할인 대상: 5(오)인 이상의 수강생에 대하여 10(십)% 할인

## 현재 일정

Windows용 WDF 드라이버 작성 -- 액셀레이티드 에디션

- 대한민국, 서울: 2010년 4월 26일
- 중국, 상하이: 2010년 5월 10일
- 인도, 하이데라바드: 2010년 5월 24일

## 세미나 대상

KMDF를 사용하여 Windows 드라이버를 설계, 개발 또는 시험하는 방법의 이해가 필요한 엔지니어. WDM을 사용하여 Windows 드라이버를 작성하는 방법을 알고 있는 엔지니어에게는 KMDF를 통하여 Windows 드라이버를 간편하게 작성하는 방법 교육. 이 세미나 이후에, Windows용 드라이버를 개발한 적이 없는 엔지니어는 왜 사람들이 Windows 드라이버 작성이 어렵다고 이야기하는지에 대하여 의아하게 됨(OSR을 통하여 KMDF 사용 방법을 이해하면, 사람들이 말하는 것처럼 어렵지 않다는 사실을 발견하게 됨!)

유의사항: 개최 장소와 상관없이, 이 세미나는 영어로 진행됨.

## 참가자 조건

이 세미나에 참여하는 수강생은 운영체계에 대한 견고한 일반 지식(사용자 모드 vs 커널 모드, 가상 메모리 개념, 동치 처리 이슈), 장치 개념(레지스터, 인터럽트), Windows 운영체제 구조의 기본개념에 대한 이해가 요구된다. 장치 드라이버 개발에 대한 이전 경험(모든 종류의 운영체제)을 가지고 있는 경우, 세미나를 이해하는 데에 커다란 도움이 된다.

## 세미나 개요

- **도입**  
환영사, 세미나 목표 및 목적과 WDF와 KMDF에 대한 간략 소개
- **Windows 아키텍처 개요**  
Windows 운영체제 아키텍처의 간략 검토, KMDF 드라이버 개발자에 필요한 세부정보에 대한 특히 중점 소개.
- **WDF 개체 모델**  
WDF 개체 특징과 분류. KMDF에서 개체의 인스턴스화 및 사용 방법. 가장 일반적인 WDF 개체에 대한 개요.
- **드라이버 초기화**  
KMDF 드라이버와 관련 장치의 초기화 방법. 장치 부착, 전원 증가, 전원 감소 등의 일반적인 플러그 앤 플레이(PnP)와 전원관리 이벤트의 처리. KMDF 드라이버 알림 방법과 하드웨어 자원 요청 방법.
- **드라이버 설치**  
KMDF 드라이버용 설치 제어 파일 작성 방법. 10가지의 최다 사용 빈도 INF 파일 섹션에 대한 논의. KMDF 공동 설치 프로그램과 INF 파일에서 지정하는 방법 설명.
- **작성 및 디버깅**  
Windows 드라이버 키트(WDK)를 사용하여 KMDF 드라이버 작성. Windows 디버거(WinDbg) 설정 방법과 "in flight recorder"에서 WDF 로그 검색하기와 같은 WDF 커널 디버거 확장(WDFKD)에 대한 간략한 개요. WDF 드라이버에 적절한

다양한 드라이버 디버깅 도구 (Driver Verifier, Static Driver Verifier, Prefast for Drivers)에 대한 논의.

- **Windows 디바이스 트리**

Windows PnP 하위 시스템의 드라이버 발견 및 열거 방법에 대한 설명. 실제 장치 개체(PDO), 기능 장치 개체 (FDOs), 필터 장치에 대한 설명. 필터 드라이버의 기능 작동 방법. 요청이 처리되어 Windows I/O 하위 시스템에서 드라이버에서 드라이브로 전달되는 방법.

- **Interrupt Levels 및 지연된 프로시저 호출**

이 강의 모듈에서는 인터럽트 요청 수준(IRQL)과 다양 IRQL의 주요 중요 개념과 IRQ의 Windows 메이커별 사용 방법을 논의한다. 지연된 프로시저 호출 (DPCs)과 인터럽트 서비스 루틴 완료(DPCforISR)용으로 Windows에서 사용되는 방법을 논의한다.

- **큐와 요청**

이 섹션에서는 WDFQUEUE와 WDFREQUESTS를 다룬다. 이 섹션의 주제에는 큐가 인스턴스화하는 방법, 큐 디스패치 유형, 요청을 정렬하는 데에 큐가 사용되는 방법 등을 포함된다. 프레임워크 요청과 요청이 처리되어 완료되는 방법을 논의한다.

- **I/O 대상**

로컬, 원격 및 특수 I/O 대상이 논의된다. How 요청을 동기 및 비동기적으로 시스템의 다른 드라이버에 처리하기 위해 전달하는 방법을 논의한다. 완료 루틴도 다루어 진다.

- **버퍼 메서드와 IOCTL 정의**

이 섹션에서는 요청자 데이터 버퍼의 다양한 설명 방법이 논의된다. 직접 I/O, 버퍼 I/O와 "Neither I/O"가 설명, 비교, 대비된다. 사용자 장치 IO 제어 코드 (IOCTL) 정의 방법과 이미 설명된 버퍼링 메서드가 IOCTL가 적용되는 방법도 논의된다.

- **직렬화**

이 강의 모듈에서는, 자신의 드라이버 안의 공유 데이터에 대한 액세스 동기화에 관련된 문제를 논의한다. 잘못 이해되고 있는 KMDF 동기화 범위에 대하여 전면적으로 설명하고, 자동 직렬화 매개 변수를 통하여 동기화 범위를 다른 콜백

루틴으로 확장에 대하여 설명한다. WDFSPINLOCK와 WDFWAITLOCK도 각각이 구현되는 기본 개념과 사용 방법에 대하여 논의된다.

- **USB 개념**

장치, 구성, 인터페이스 설명자 등의 USB 기본 개념이 논의된다. 엔드포인트와 파이프가 설명된다.

- **WDF USB 드라이버 구현하기**

이 섹션에서는 USB 드라이버가 KMDF에서 구현되는 방법을 설명한다. 이 섹션에는 구성과 인터페이스가 선택되는 방법 및 엔드포인트가 검색되는 방법도 포함한다. Endpoint 0에 벤더 명령어 보내는 방법. 벌크 엔드포인트 및 인터럽트 엔드포인트 사용하기. 선택적 절전 모드(USB 장치 전원 관리)를 지원하는 WDF 연속 리더가 간략하게 논의된다.

- **프로그램식 I/O 장치용 WDF 드라이버 구현**

이 섹션에서는 요청을 수신 및 처리에 대한 논의를 드라이버가 프로그램식 I/O 형 PCI와 PCI Express 장치에 대한 요청을 처리하는 방법으로 확장한다. 하나의 KMDF 드라이버 안에서 자신의 인터럽트 서비스 루틴(ISR)과 IRS 완료에 대한 지연된 프로시저 호출을 구현하는 방법을 설명한다.

- **정리, 달기 및 취소**

일반적으로 정리 및 달기 과정의 부분으로 발생하는 프로세스로서 큐에 있는 요청과 진행 중 요청을 처리하는 전략에 대하여 논의한다.

- **유용한 클래스**

WDFCOLLECTION, WDFWORKITEM, WDFTIMER과 같은 몇 가지 프레임워크 클래스에 대한 간략한 설명이 제공하여, 자신이 가진 "노하우"를 향상시킬 수 있게 한다.