บทที่ 9
ไดเร็กโชว์ (DirectShow)

9.1 แนะนำไดเร็กโชว์

Microsoft® DirectShow® Application Programming Interface (API) คือ สถาปัตยกรรมของมีเดียสตรีมมิ่ง (media-streaming) สำหรับระบบปฏิบัติการวินโดว์ การใช้ไดเร็กโชว์ (DirectShow) ช่วยให้แอพพลิเคชันสามารถเล่นไฟล์วีดีโอคุณภาพสูงและเล่นหรือบันทึกไฟล์เสียงได้

ไดเร็กโชว์สนับสนุนไฟล์รูปแบบต่างๆ จำนวนมาก เช่น Advanced Streaming Format (ASF), Motion Picture Experts Group (MPEG), Audio-Video Interleaved (AVI), MPEG Audio Layer-3 (MP3) และ Wave รวมทั้งสนับสนุนการบันทึกโดยผ่านทางอุปกรณ์ Windows Driver Model (WDM) หรืออุปกรณ์อื่นๆ สำหรับวินโดว์ ไดเร็กโชว์ได้รวมเอาเทคโนโลยีของไดเร็กเอ็กซ์อื่นๆ มาไว้รวมกัน โดยจะตรวจสอบโดยอัตโนมัติและเลือกใช้ฮาร์ดแวร์ในการเร่งความเร็ววีดีโอและออดิโอ (video and audio acceleration hardware) เมื่อมีอุปกรณ์นั้นอยู่ นอกจากนั้นยังสนับสนุนการใช้งานโดยไม่ผ่านฮาร์ดแวร์อีกด้วย

ไดเร็กโชว์ทำให้การเล่นไฟล์มีเดีย, การแปลงรูปแบบไฟล์ และการบันทึกเป็นเรื่องง่าย แต่ในขณะเดียวกันให้สามารถเข้าถึงสถาปัตยกรรมควมคุมสตรีมที่อยู่ข้างใต้ได้เช่นกันสำหรับแอพพลิเคชันที่ต้องการผลลัพธ์ที่ต่างออกไป นั่นคือ สามารถที่จะสร้างองค์ประกอบของไดเร็กโชว์ขึ้นมาใหม่ได้เพื่อที่จะสนับสนุนไฟล์รูปแบบใหม่ๆ หรือเอ็ฟเฟ็คที่แตกต่างออกไป

ไดเร็กโชว์อยู่บนพื้นฐานของ Component Object Model (COM) ในการเขียนแอพพลิเคชันที่ใช้งานไดเร็กโชว์จึงมีความจำเป็นที่ต้องทำความเข้าใจกับ COM client programming (การเขียนแอพพลิเคชันที่ทำงานบนเครื่องคอมพิวเตอร์เดียวโดยใช้ COM) ซึ่งส่วนมากแล้วมักจะไม่จำเป็นที่จะต้องสร้าง COM object ขึ้นมาเอง เนื่องจากไดเร็กโชว์ได้สร้างองค์ประกอบที่ต้องการไว้ให้แล้ว (แต่สามารถสร้างเพิ่มเติมได้)

9.2 เริ่มต้นการเขียนโปรแกรมด้วยไดเร็กโชว์

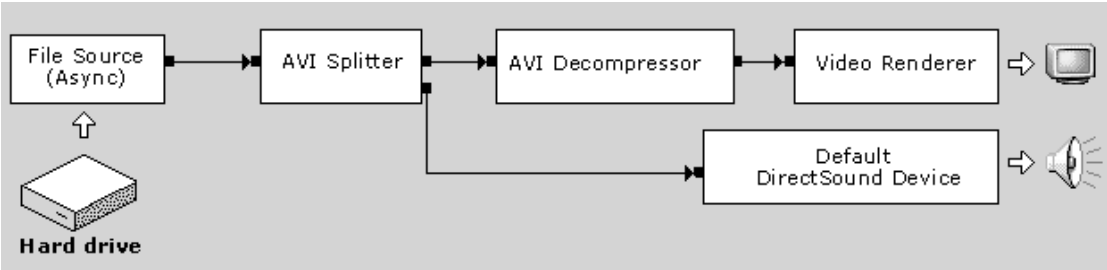
ในการเริ่มต้นเขียนโปรแกรมด้วยไดเร็กโชว์จำเป็นที่จะต้องทำความเข้าใจกับศัพท์และองค์ประกอบที่ประกอบกันในการทำงานของไดเร็กโชว์ดังนี้

ฟิลเตอร์ (Filter), ฟิลเตอร์กราฟ (Filter Graphs) และ ฟิลเตอร์กราฟเมเนเจอร์ (Filter Graph Manager)

ในไดเร็กโชว์มีองค์ประกอบของซอฟท์แวร์ (Software component) ที่เรียกว่า ฟิลเตอร์ (filter) เป็นองค์ประกอบพื้นฐาน ซึ่งฟิลเตอร์จะทำกระบวนการ 1 อย่างกับมัลติมีเดียสตรีม เช่น มีไดเร็กโชว์ฟิลเตอร์ที่ทำการอ่านไฟล์, รับข้อมูลจากกล้องวีดีโอ, ถอดรหัส (decode) บางส่วนของสตรีม เช่น MPEG-1 หรือ ส่งข้อมูลไปยังกราฟิกการ์ด

ฟิลเตอร์รับข้อมูลและสร้างเอาท์พุตออกมา เช่น ถ้าฟิลเตอร์ถอดรหัส MPEG-1 video อินพุตคือสตรีมที่ผ่านการเข้ารหัสด้วย MPEG และเอาท์พุตคือ สตรีมข้อมูลที่ไม่มีการบีบอัดในแบบ RGB video ดังนั้นในการทำงาน 1 อย่าง แอพพลิเคชันจะต้องนำเอาฟิลเตอร์เหล่านี้มาต่อเข้าด้วยกันโดยนำเอาท์พุตของ

ฟิลเตอร์หนึ่งมาต่อกับอินพุตของอีกฟิลเตอร์หนึ่ง กลุ่มของการเชื่อมต่อของฟิลเตอร์นี้เรียกว่า ฟิลเตอร์กราฟ ดังรูป



รูปที่ 9-1 แสดงการเชื่อมต่อฟิลเตอร์เพื่อเล่นไฟล์ AVI

แอพพลิเคชันไม่จำเป็นที่จะต้องเข้ามาควบคุมแต่ละฟิลเตอร์โดยตรง เนื่องจากไดเร็กโชว์ได้เตรียมองค์ประกอบขั้นสูง (high-level component) ในการควบคุมไว้แล้ว ที่เรียกว่า ฟิลเตอร์กราฟเมเนเจอร์ (Filter Graph Manager)

ฟิลเตอร์กราฟเมเนเจอร์ควบคุมการไหลของสตรีมข้อมูล โดยแอพพลิเคชันสามารถควบคุมได้โดยการเรียกฟังก์ชันง่ายๆ เช่น Run (ส่งข้อมูลเข้าสู่กราฟ), Stop (หยุดการไหลของข้อมูล) ถ้าต้องการที่จะควบคุมฟิลเตอร์โดยตรงก็สามารถทำได้เช่นกันโดยผ่านทาง COM Interface นอกจากนั้น ฟิลเตอร์กราฟเมเนเจอร์ยังสามารถส่งค่าแจ้งเหตุการณ์ (event notification) ให้กับแอพพลิเคชันได้อีกด้วย ตัวอย่าง เช่น การแจ้งว่าจบสตรีมข้อมูลแล้ว เป็นต้น

นอกจากนั้น ฟิลเตอร์กราฟเมเนเจอร์ยังทำให้กระบวนการในการสร้างฟิลเตอร์กราฟมีความง่ายดายยิ่งขึ้น เช่น เพียงแค่บอกชื่อของไฟล์ที่ต้องการจะเล่น ฟิลเตอร์กราฟจะไปสร้างกราฟที่เหมาะสมสำหรับการเล่นไฟล์ให้ทันที

ฟิลเตอร์ในไดเร็กโชว์แบ่งเป็นประเภทต่างๆ ดังนี้

Source Filter

แทนข้อมูลมัลติมีเดียที่เป็นวัตถุดิบในกระบวนการ ซึ่งอาจจะได้มาจากไฟล์บนฮาร์ดดิสต์ หรือจากแผ่น CD, DVD ในไดร์ฟ หรืออาจจะได้มาจากรายการสดผ่านทางอุปกรณ์รับสัญญาณโทรทัศน์ หรือจากกล้องดิจิตอลเป็นต้น

Transform Filter

เป็นฟิลเตอร์ที่รับข้อมูลวัตถุดิบหรือข้อมูลที่มีการประมวลผลแล้วบางส่วนมาเพื่อที่จะประมวลผลข้อมูลเหล่านั้น ซึ่งการประมวลผลอาจจะเป็นการแบ่งข้อมูลออกเป็นส่วนย่อยๆ , การบีบอัดและขยาย และการแปลงรูปแบบ (format converter)

Render Filter

เป็นฟิลเตอร์ที่จะรับข้อมูลที่ได้ประมวลผลแล้วมาเพื่อแสดงภาพบนจอมอนิเตอร์หรือส่งสัญญาณเสียงผ่านทางลำโพง หรืออุปกรณ์อื่นๆ รวมทั้งการเขียนลงไฟล์บนดิสต์ ที่มีชื่อเรียกเฉพาะว่า file writer

Pins

จากรูปฟิลเตอร์กราฟจะเห็นว่าข้อมูลสตรีมมิ่งจะเดินทางจากซอร์สฟิลเตอร์ผ่านฟิลเตอร์อื่นๆ และไปจบที่เรนเดอร์ฟิลเตอร์ การเชื่อมต่อระหว่างฟิลเตอร์จะต้องใช้พินในการเชื่อมต่อ พินคือ รายละเอียดในระดับล่างของข้อมูลที่จะส่งระหว่างฟิลเตอร์ พินเป็น COM object ที่สนับสนุนโดย Ipin COM interface มีทิศทาง (อินพุตและเอาท์พุต) และมีการติดต่อกับบางส่วนของฟิลเตอร์ที่มีอยู่ในกราฟ พินแสดงการเชื่อมต่อระหว่างแต่ละฟิลเตอร์ พินจะรู้ว่าชนิดของมีเดียใดที่ใช้งานได้และจะมีการเจรจากันระหว่างพินเพื่อเลือกชนิดของมีเดียที่จะใช้ในการติดต่อ เมื่อกำหนดชนิดได้แล้วจะต้องมีการเจรจาต่อไปในการกำหนดวิธีการในการส่งข้อมูลระหว่างฟิลเตอร์

มีเดียแซมเพิล (Media Samples)

หลังจากข้อมูลวัตถุดิบได้ส่งผ่านเข้าสู่กราฟไม่ว่าจะมาจากไหนก็ตาม จะต้องมีการจัดให้เป็นหน่วยที่มีความหมาย (meaningful unit) ที่เรียกว่า มีเดียแซมเพิล (Media Sample) ในไดเร็กโชว์มีเดียแซมเพิลถูกห่อหุ้ม (Wrap) ด้วย COM object ที่อิมพลีเมนต์ (Implement) ImediaSample2 ในวีดีโอคือ ข้อมูล 1 เฟรมเป็นต้น

นาฬิกา (Clock)

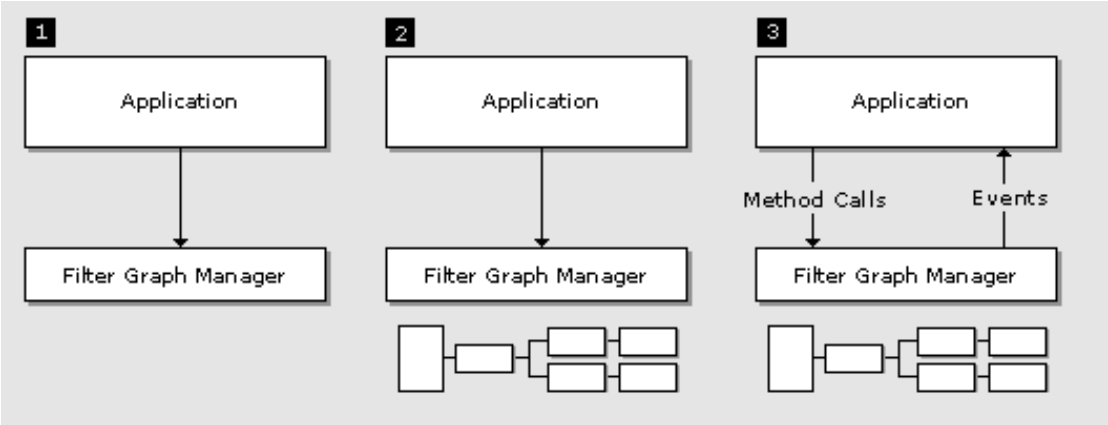
ในกระบวนการที่เกี่ยวข้องกับมัลติมีเดีย จะต้องมีการควบคุมจังหวะของการแสดงผลให้มีความถูกต้อง เช่น วีดีโอจะต้องควบตุมอัตราเฟรม (Frame rate) ให้เป็นไปตามที่กำหนด ซึ่งไดเร็กโชว์มีนาฬิกาจะต้องใช้งานร่วมกันในกระบวนการต่างๆ โดยฟิลเตอร์กราฟเมเนเจอร์จะเลือกนาฬิกาขึ้นมา และสั่งให้ฟิลเตอร์ทุกตัวใช้นาฬิกาตัวนั้นเป็นจังหวะเดียวกัน

ตัวจองหน่วยความจำ (Allocator)

มีฟิลเตอร์ 2 ตัวมีการเชื่อมต่อกัน พินจะต้องตกลงในรายละเอียดในการส่งข้อมูลและชนิดของข้อมูล ในการเชื่อมต่อจะต้องกำหนดขนาด, ตำแหน่ง และจำนวนของแซมเพิลที่จะใช้ ขนาดของแซมเพิลจะขึ้นอยู่กับชนิดของมีเดียและรูปแบบของมีเดีย และตำแหน่งของบัฟเฟอร์อาจอยู่บนเมมโมรีของระบบหรือที่ใดๆก็ได้ เช่น บนการ์ดบันทึกภาพวีดีโอ การสร้างและจัดการสิ่งเหล่านี้ทำโดยตัวจองหน่วยความจำ

9.3 การเขียนโปรแกรมโดยใช้ไดเร็กโชว์

โดยปกติแล้วแอพพลิเคชันที่จะใช้งานไดเร็กโชว์จะมีขั้นตอนพื้นฐาน 3 ขั้น ดังรูป



รูปที่ 9-2 ขั้นตอนการสร้างไดเร็กโชว์

ขั้นที่ 1 การสร้างอินสแตนท์ (Instance) ของฟิลเตอร์กราฟเมเนเจอร์ โดยเรียกฟังก์ชัน CoCreateInstance

ขั้นที่ 2 ใช้ฟิลเตอร์กราฟเมเนเจอร์ในการสร้างฟิลเตอร์กราฟ (อาจจะต้องใช้องค์ประกอบอื่นๆที่มีอยู่ในไดเร็กโชว์ช่วย)

ขั้นที่ 3 ควบคุมฟิลเตอร์กราฟและตอบสนองต่อเหตุการณ์ (event)

9.4 การเล่นไฟล์ (Play a File)

การที่จะเล่นไฟล์โดยใช้ไดเร็กโชว์มีขั้นตอนดังนี้

1. สร้างอินสแตนท์ของฟิลเตอร์กราฟเมเนเจอร์
2. ใช้ฟิลเตอร์กราฟเมเนเจอร์เพื่อสร้างฟิลเตอร์กราฟ
3. ใช้ฟิลเตอร์กราฟเมเนเจอร์ในการทำงานของฟิลเตอร์กราฟ
4. รอให้การเล่น (playback) จบ

การจะทำขั้นตอนทั้ง 4 อย่างข้างต้นจะต้องใช้ COM Interface ต่อไปนี้

IGraphBuilder สร้างฟิลเตอร์กราฟ

IMediaControl จัดการมีเดียสตรีมมิ่งในฟิลเตอร์กราฟ

ImediaEvent จัดการเหตุการณ์ของฟิลเตอร์กราฟ

ซึ่งอินเทอร์เฟสข้างต้นทั้งหมดคือ ฟิลเตอร์กราฟเมเนเจอร์

ขั้นที่ 1

เรียกฟังก์ชัน CoInitialize เพื่อตั้งค่าเริ่มต้นให้กับ COM Library แล้วเรียก CoCreateInstance เพื่อสร้างฟิลเตอร์กราฟเมเนเจอร์ ดังตัวอย่าง

```
IGraphBuilder *pGraph;
CoInitialize(NULL);
CoCreateInstance(CLSID_FilterGraph, NULL, CLSCTX_INPROC_SERVER,
        IID_IGraphBuilder, (void **)&pGraph);
```

ฟังก์ชัน CoCreateInstance จะคือค่าพอยเตอร์ที่ชี่ไปยังอินเทอร์เฟส IgraphBuiler ในฟิลเตอร์กราฟเมเนเจอร์คืนมาให้ แล้วใช้อินเทอร์เฟสนี้ในการขออีก 2 อินเทอร์เฟสที่เหลือดังตัวอย่าง

```
IMediaControl *pMediaControl;
IMediaEvent   *pEvent;
pGraph->QueryInterface(IID_IMediaControl, (void **)&pMediaControl);
pGraph->QueryInterface(IID_IMediaEvent, (void **)&pEvent);
```

ขั้นที่ 2

ใช้เมทอด IgraphBuiler::RenderFile เพื่อสร้างฟิลเตอร์กราฟที่จะใช้ในการเล่นไฟล์นั้นๆ ค่าที่ส่งให้เมทอดนี้คือ ชื่อของไฟล์ที่จะเล่นโดยจะต้องอยู่ในรูปแบบ wide character (2-byte) Unicode™ string โดยใช้ “L” นำหน้าชื่อไฟล์ที่ต้องการ (ทำการแปลง ASCII เป็น wide character string)

```
pGraph->RenderFile(L"C:\\Hello_World.avi", NULL);
```

ในกรณีที่ไม่พบไฟล์หรือไม่สามารถสร้างฟิลเตอร์กราฟได้เมทอดจะส่งค่า VFW_E_NOT_FOUND คืนมาให้

ขั้นที่ 3

หลังจากฟิลเตอร์กราฟเมเนเจอร์สร้างฟิลเตอร์กราฟแล้วจึงเริ่มกระบวนการเล่นไฟล์โดยการเรียกเมทอด ImediaControl::Run เพื่อให้กราฟเริ่มทำงาน โดยจะส่งข้อมูลเข้าสู่กราฟ

ขั้นที่ 4

เมทอด ImediaEvent::WaitForCompletion จะบล็อกจนกว่าไฟล์จะเล่นจบ เนื่องจากการเล่นไฟล์จะเกิดขึ้นในเทรด (thread) ที่สร้างขึ้นต่างหาก