

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมตัดต่อภาพถ่ายวิดีโอเพื่อใช้สำหรับการตัดต่อวิดีโอส่วนบุคคล
VIDEO EDITOR FOR MANIPULATING PERSONAL VIDEOS OR MOVIES



นายสรุต เรียวรักษา

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

พ.
6292/2
2545

เลขหมู่.....
เลขทะเบียน... 49999
วัน,เดือน,ปี 16 เม.ย. 2547

b.....
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2545/49999

โปรแกรมตัดต่อภาพถ่ายวิดีโอเพื่อใช้สำหรับการตัดต่อวิดีโอส่วนบุคคล
VIDEO EDITOR FOR MANIPULATING PERSONAL VIDEOS OR MOVIES



โดย

นายศรุต เรียวรักษา

อาจารย์ที่ปรึกษา

ดร. อรณิตร จิตต์โสภิรักษ์

ปริญญาานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

ปริญญาานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมตัดต่อภาพถ่ายวิดีโอเพื่อใช้สำหรับการตัดต่อวิดีโอส่วนบุคคล

VIDEO EDITOR FOR MANIPULATING PERSONAL VIDEOS OR MOVIES

ผู้จัดทำ

1. นายศรุต เรียวรักษา รหัสประจำตัว 42010342

อรุณทิพย์ วัฒน. อาจารย์ที่ปรึกษา
(ดร. อรณัฏฐ์ จิตต์โสภาคย์)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมตัดต่อภาพถ่ายวิดีโอเพื่อใช้สำหรับการตัดต่อวิดีโอส่วนบุคคล

นายศรุต เรียวรักษา 42010342

ดร. อรรถธร จิตต์โสภักตร์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2545

บทคัดย่อ

เมื่อการพัฒนาทางดิจิทัล มัลติมีเดีย (Digital Multimedia) มีความก้าวหน้าไปมาก ประกอบกับความต้องการในการที่จะใช้งานมัลติมีเดียให้มีความสะดวกมากยิ่งขึ้น เป็นสิ่งจูงใจให้ผู้ผลิตทำการพัฒนาผลิตภัณฑ์สู่ตลาดจำนวนมาก ในการพัฒนาเหล่านั้นสิ่งที่ถูกพัฒนาอย่างต่อเนื่องก็คือ ระเบียบวิธีการในการประมวลผลรหัสดิจิทัล ซึ่งได้มีผู้คิดค้นวิธีการใหม่ๆ จำนวนมาก จึงทำให้เกิดแนวความคิดในการสร้างมาตรฐานเพื่อความเข้ากันได้ในการพัฒนาระบบประมวลผลวิดีโอโดยจัดตั้งกลุ่มที่เรียกว่า MPEG (Moving Picture Experts Group)

โครงการนี้ได้นำหลักการของไดเรกโชว์ เอพีไอ (Directshow API) มาใช้ในการพัฒนาโปรแกรมตัดต่อวิดีโอประเภท MPEG-2 ให้โปรแกรมหักความสามารถในการเล่นและเก็บภาพ (snap shot) รวมทั้งตัดต่อวิดีโอได้ โดยโครงการนี้เป็นโปรแกรมที่เขียนด้วยภาษาวิซวลเบสิก (Visual Basic) ซึ่งสามารถทำงานร่วมกับวิดีโอประเภท MPEG ได้

Video Editor For Manipulating Personal Videos Or Movies

Sarut Reoraksa

Dr. Orachat Chitsopak Advisor

ABSTRACT

In recent year, the digital multimedia has great improvement. The sharply grown market convince manufactures to advantage their product for competition. Simultaneously, they develop many new digital data processing format and algorithm. For compatibility, a group named “MPEG” (Moving Picture Expert Group) was founded.

This project is implemented by DirectShow API that supported MPEG-2 video editing project. The program can also playback, capture and edit the video clip. This project brings the advantage of DirectShow API to use with MPEG2 video editing project. It brings more functionality to playback capture and edit video clip. Visual Basic is the language used and compatible with MPEG file. Program that created in this project is using Microsoft Visual Basic. It can cooperate with MPEG file.

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญตาราง	VI
สารบัญรูปภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนการดำเนินงาน	2
1.6 เนื้อหาของปริญญานิพนธ์	2
บทที่ 2 ไคเรกโซว์ เอฟไอ	3
2.1 สถาปัตยกรรมของไคเรกโซว์	3
2.2 สถาปัตยกรรมของไคเรกโซว์ฟิลเตอร์	4
2.3 การใช้ไคเรกโซว์	6
2.4 การใช้ไคเรกโซว์ กับวิซวลเบลติก	6
2.4.1 การสร้างฟิลเตอร์โดยใช้วิซวลเบลติก	6
2.4.1.1 การสร้างฟิลเตอร์ กราฟ	6
2.4.1.2 การสร้างฟิลเตอร์ กราฟแบบสมบรูณ์	7
2.4.1.3 สรุปการใช้ข้อบ่งชี้ของไคเรกโซว์ คุณสมบัติและเมธอด	7
2.4.2 การควบคุมฟิลเตอร์ กราฟโดยใช้วิซวลเบลติก	8
2.4.2.1 ไคเรกโซว์ฟิลเตอร์และฟิลเตอร์ กราฟ	8
2.4.2.2 การควบคุมฟิลเตอร์ กราฟในวิซวลเบลติก	10
2.4.2.2.1 การเตรียมใช้ออบเจกต์ของไคเรกโซว์	10
2.4.2.2.2 การ Instantiating ฟิลเตอร์ กราฟ	10
2.4.2.2.3 การเรนเดอร์วีดีโอ	11
2.4.2.2.4 การควบคุมเสียง	12
2.4.2.2.5 การปรับเปลี่ยนสเกลวีดีโอเอาต์พุต	13
2.4.2.2.6 การติดตามสถานะ	14
2.4.2.2.7 การรับและตั้งค่าตำแหน่งเริ่มต้น	14
2.4.2.2.8 การรับและตั้งค่าอัตราเร็ว	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
2.4.2.2.9 Cleaning up	15
2.4.2 ออบเจ็กต์ในวิซวลเบสิก	15
บทที่ 3 DES (DirectShow Editing Services)	17
3.1 DirectShow Editing Services	17
3.1.1 DirectShow Editing Services เบื้องต้น	17
3.1.2 หลักการพื้นฐานของ DES	18
3.1.2.1 DES และ เอพีไอหลักของ DirectShow	18
3.1.3 Timeline Model	18
3.1.3.1 โครงสร้างของ Timeline	18
3.1.3.2 จะเรนเดอร์ Timeline ได้อย่างไร	19
3.1.4 เวลาที่เกี่ยวข้องใน DES	20
3.1.4.1 Time Value	20
3.1.4.2 Frame Rates	21
3.2 ออบเจ็กต์ที่ใช้งานตัดต่อวิดีโอของ DES	22
บทที่ 4 การออกแบบ โปรแกรม	24
4.1 หลักการและแนวคิดการออกแบบ	24
4.2 การทำงานของอินเทอร์เฟซและออบเจ็กต์ที่สำคัญ	24
4.2.1 ออบเจ็กต์ MediaDet	24
4.2.2 ออบเจ็กต์ IBasicAudio	25
4.2.3 ออบเจ็กต์ IBasicVideo	25
4.2.4 ออบเจ็กต์ IMediaEvent	25
4.2.5 ออบเจ็กต์ IVideoWindow	25
4.2.6 อินเทอร์เฟซ IMediaControl	26
4.2.7 ออบเจ็กต์ IMediaPosition	26
4.2.8 ออบเจ็กต์ FilgraphManager	26
4.3 การทำงานของออบเจ็กต์ที่สำคัญที่ใช้ในการตัดต่อ	26
4.3.1 ออบเจ็กต์ AMTimeline	26
4.3.2 ออบเจ็กต์ AMTimelineObj	27
4.3.3 ออบเจ็กต์ AMTimelineSrc	27
4.3.4 ออบเจ็กต์ AMTimelineTrack	27
4.3.5 ออบเจ็กต์ AMTimelineGroup	27
4.3.6 ออบเจ็กต์ AMTimelineComp	28
4.3.7 ออบเจ็กต์ SmartRenderEngine	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
บทที่ 5 การทดลองและผลการทดลอง	30
5.1 ความต้องการของระบบ	30
5.2 ระบบที่ใช้ทดสอบ	30
5.3 การทดสอบโปรแกรม	30
บทที่ 6 สรุปและวิจารณ์	34
6.1 บทวิจารณ์	34
6.2 แนวทางในการพัฒนาโปรแกรม	34
6.3 บทสรุป	34
บรรณานุกรม	34



สารบัญตาราง

	หน้า
ตารางที่ 2-1 แสดงออบเจกต์ คุณสมบัติและเมธอดในโคเรกโซว์	8
ตารางที่ 2-2 แสดงอินเทอร์เฟซของ Quartz.dll	9
ตารางที่ 2-3 แสดงออบเจกต์ในวิซวลเบสิก	16
ตารางที่ 4-1 แสดงการทำงานของโปรแกรม	24



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2-1 การติดต่อถึงไคลเอนท์	4
รูปที่ 2-2 ฟิวเจอร์ กราฟเพื่อเล่นไฟล์ MPEG	5
รูปที่ 2-3 ฟิวเจอร์ของไฟล์ AVI	8
รูปที่ 3-1 แสดงโครงสร้างของ Timeline	19
รูปที่ 3-2 แสดงการเรนเดอร์ Timeline	20
รูปที่ 3-3 แสดงเวลาที่ใช้ในโปรเจกต์	21
รูปที่ 3-4 แสดง playback rate	22
รูปที่ 4-1 การติดต่อกับฟิวเจอร์ กราฟ	29
รูปที่ 5-1 รูปเมื่อทำการเปิดไฟล์	30
รูปที่ 5-2 เมื่อ Playback วิดีโอ	31
รูปที่ 5-3 แสดงการค้นหภาพ	31
รูปที่ 5-4 แสดงภาพที่ทำการ Capture	32
รูปที่ 5-5 ภาพขณะเล่นวิดีโอ	32
รูปที่ 5-6 แสดงวิดีโอที่ตัดออกมา	33
ภาคผนวก ก	34
บรรณานุกรม	36

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากปัจจุบันวิวัฒนาการในการบันทึกและเก็บภาพถ่ายวิดีโอก้าวหน้าไปมาก วิดีโอส่วนใหญ่จะอยู่ในรูปแบบซึ่งถูกบีบอัดข้อมูลเรียบร้อยแล้ว เพื่อที่จะลดขนาดของสัญญาณวิดีโอให้ใช้เนื้อที่ในหน่วยความจำน้อยลงในการเก็บข้อมูล รวมทั้งทำให้สามารถรับส่งสัญญาณวิดีโอได้อย่างรวดเร็ว นอกจากการเก็บและรับส่งสัญญาณวิดีโอแล้ว ผู้ใช้ยังมีความต้องการที่จะเปิดดูสัญญาณภาพถ่ายวิดีโอ นั้น และยังต้องการที่จะสามารถตัดบางช่วงของสัญญาณวิดีโอออก เพิ่มเติมคลิปวิดีโอใหม่เข้าไปในสัญญาณวิดีโอที่มีอยู่เดิม หรือเลือกภาพบางภาพออกมาแยกเก็บไว้ เช่น ฟังก์ชันดิงภาพ ถึงแม้ว่าจะมีโปรแกรมที่ใช้ทางการค้าอยู่บ้างแล้ว แต่เนื่องจากข้อจำกัดในด้านราคาลิขสิทธิ์ ทำให้การนำมาใช้และการเปลี่ยนแปลงเพิ่มเติมฟังก์ชันใหม่ที่ต้องการเป็นไปได้ยาก ดังนั้นจึงมีแนวคิดที่จะสร้างโปรแกรมที่สามารถ ตัดต่อภาพถ่ายวิดีโอขึ้นมาใช้เอง โดยโปรแกรมนี้จะต้องสามารถถอดสัญญาณวิดีโอที่อยู่ในรูป MPEG-2 เพื่อแสดงภาพถ่ายวิดีโอทางจอภาพ นอกจากนี้ยังสามารถตัดสัญญาณวิดีโอบางช่วงออกทิ้งไปหรือบันทึกเก็บไว้ในรูป MPEG-2 หรือแทรกคลิปวิดีโอที่ได้รับมาจากผู้ใช้ เข้าไปในสัญญาณภาพถ่ายวิดีโอที่มีอยู่แล้วได้ ทั้งนี้ยังสามารถเก็บภาพในสัญญาณภาพถ่ายวิดีโอออกมาเพื่อแยกไว้ในรูปของการบีบอัดข้อมูลในรูปแบบต่างๆได้

โครงการนี้ได้นำโคเรคโซว์ เอพีไอมาใช้เนื่องจากมีสถาปัตยกรรมในระดับโครงสร้างข้อมูลบนแพลตฟอร์มวินโดวส์ และยังรองรับข้อมูลหลายรูปแบบอีกด้วยซึ่งให้ผลงานที่มีคุณภาพดี โดยนำมาใช้กับการเล่นไฟล์วิดีโอ ในส่วนของการตัดต่อได้เลือกใช้ DES ซึ่งเป็นเอพีไอที่สนับสนุนการตัดต่อวิดีโอมาใช้ในการพัฒนา และเขียนโปรแกรมด้วยภาษาวิซวลเบสิก

1.2 วัตถุประสงค์ของปฏิญานิพนธ์

เพื่อศึกษาการใช้โคเรคโซว์ เอพีไอและสามารถนำความรู้ที่ได้ศึกษามาสร้างโปรแกรมสำหรับการตัดต่อภาพถ่ายวิดีโอส่วนบุคคลโดยสามารถใช้กับไฟล์วิดีโอ MPEG-2 ได้ โปรแกรมมีการใช้งานง่ายไม่ซับซ้อน ซึ่งโปรแกรมจะมีความสามารถคือ ตัดและเล่นไฟล์วิดีโอได้

1.3 ขอบเขตของปฏิญานิพนธ์

- ศึกษาการเล่น และตัดต่อไฟล์วิดีโอ
- ศึกษาการเขียนโปรแกรมด้วยโคเรคโซว์
- ศึกษาโครงสร้างของ MPEG-2
- ศึกษาการเขียนโปรแกรมด้วยภาษาวิซวลเบสิก

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- ความรู้ความเข้าใจเกี่ยวกับโคเรกโซว์
- ความรู้ความเข้าใจในเรื่องโครงสร้างของ MPEG-2
- สามารถเขียนโปรแกรมโดยใช้โคเรกโซว์ได้
- สามารถเขียนโปรแกรมที่สามารถตัดต่อภาพถ่ายวิดีโอซึ่งอยู่ในรูปแบบของ MPEG-2 ณ ตำแหน่งที่ผู้ใช้ต้องการ

1.5 ขั้นตอนการดำเนินงาน

โครงการนี้มีขั้นตอนการดำเนินงานโดยเริ่มจาก

1. ศึกษาโครงสร้างของไฟล์วิดีโอ MPEG-2 ซึ่งประกอบไปด้วยส่วนต่างๆ หลายส่วน
2. ศึกษาการเขียนโปรแกรมกับโคเรกโซว์รวมทั้งศึกษาสถาปัตยกรรมของโคเรกโซว์ด้วย
3. ศึกษาการเขียนโปรแกรมด้วยภาษาวิซวลเบสิก
4. เขียนโปรแกรมโดยเริ่มจากส่วนของการเล่นวิดีโอ และส่วนของการจับภาพวิดีโอและส่วนของการตัดต่อวิดีโอ

1.6 เนื้อหาของปริญญานิพนธ์

โครงการนี้ใช้ภาษาวิซวลเบสิก ในการเขียนโปรแกรมร่วมกับการใช้โคเรกโซว์เพื่อช่วยในส่วนของการจัดการกับกระแสข้อมูลของไฟล์วิดีโอ โดยใช้ส่วนประกอบในระดับ high-level ที่เรียกว่า Filter Graph Manager ซึ่งควบคุมข้อมูลผ่าน COM (Component Object Model) ออบเจกต์ซึ่งมีอินเทอร์เฟซ (Interface) มากมายให้ใช้งาน โดยต้องอาศัยตัวถอดรหัส MPEG-2 ที่หามาได้ นอกจากนี้ได้นำ DES ซึ่งเป็นเอพีไอที่สนับสนุนในส่วนของการตัดต่อวิดีโอมาใช้

บทที่ 2

ไดเรกโทรว์ เอพีไอ (DirectShow API)

ไมโครซอฟต์ไดเรกโทรว์ เอพีไอ (Microsoft DirectShow API) เป็นสถาปัตยกรรมสำหรับสื่อข้อมูลบนแพลตฟอร์มไมโครซอฟต์วินโดวส์ (Microsoft Windows Platform) โดยเป็นส่วนหนึ่งของไมโครซอฟต์ไดเรกเอ็กซ์ เอพีไอ (Microsoft DirectX API) ซึ่งสามารถนำไดเรกโทรว์มาใช้ในการเขียนโปรแกรมเพื่อเล่นหรือเก็บภาพจากสื่อมัลติมีเดียได้และมีคุณภาพดีด้วย และยังรองรับรูปแบบของสื่อข้อมูลได้อย่างหลากหลายรวมถึงประเภท MPEG วิดีโอ เราสามารถเขียนแอปพลิเคชัน (application) โดยใช้ไดเรกโทรว์ เช่น โปรแกรมเล่นหรือตัดต่อวิดีโอได้

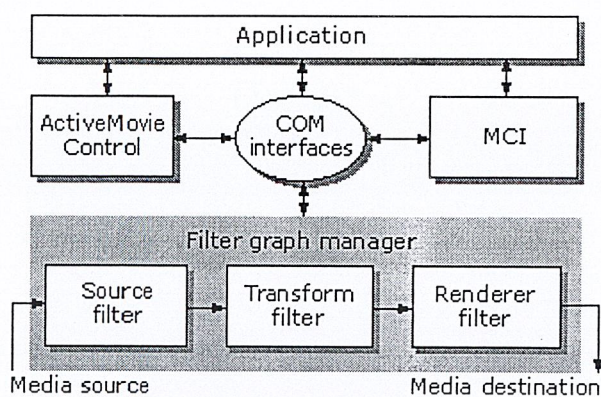
ไดเรกโทรว์ใช้หลักการของ COM ในการเขียนไดเรกโทรว์แอปพลิเคชัน เราต้องทำความเข้าใจการเขียนโปรแกรมแบบ COM client โดยที่เราไม่จำเป็นต้องสร้างออบเจกต์ COM ขึ้นมาเองไดเรกโทรว์จะกำหนดคอมโพเนนต์ที่จำเป็นมาให้

2.1 สถาปัตยกรรมของไดเรกโทรว์

สถาปัตยกรรมของไดเรกโทรว์นั้นมีการกำหนดว่าจะควบคุมและดำเนินการกับข้อมูลมัลติมีเดียอย่างไรโดยใช้ส่วนประกอบที่เรียกว่า ฟิลเตอร์ (filter) ซึ่งจะมีส่วนที่รับข้อมูลเข้าหรือออก หรือทั้งสองอย่างจากฟิลเตอร์ตัวอื่นๆ เรียกว่า pin และเมื่อนำมาเชื่อมต่อเข้าด้วยกันจะเรียกว่า ฟิลเตอร์ กราฟ (filter graph) แอปพลิเคชันจะใช้ออบเจกต์ที่เรียกว่า Filter Graph Manager เพื่อประกอบฟิลเตอร์ กราฟเข้าด้วยกันและส่งข้อมูลผ่านไปตามฟิลเตอร์ต่างๆ โดยปกติแล้ว Filter Graph Manager จะควบคุมทิศทางการไหลของข้อมูลว่าจะไปทิศทางใดโดยอัตโนมัติ ตัวอย่างเช่น จะใส่ตัว codec ให้โดยอัตโนมัติถ้าจำเป็น และยังเชื่อมต่อกับเอาต์พุตของ Transform filter ไปสู่ Renderer filter โดยอัตโนมัติอีกด้วย เราสามารถระบุฟิลเตอร์ขึ้นมาใช้งานและเชื่อมต่อกันได้เองถ้าเราไม่ต้องการใช้ฟิลเตอร์ที่มีอยู่แล้ว

Filter Graph Manager จะมีชุด COM อินเทอร์เฟซ เพื่อให้แอปพลิเคชันสามารถติดต่อฟิลเตอร์กราฟได้ แอปพลิเคชันสามารถที่จะเรียกใช้ Filter Graph Manager โดยตรงเพื่อควบคุมกระแสของข้อมูลหรือเรียกดูสิ่งที่เกิดขึ้นกับฟิลเตอร์ หรือสามารถที่จะใช้ Windows Media Player control เพื่อเล่นไฟล์มีเดีย (media)

ด้วยเหตุนี้ เราสามารถติดต่อไดเรกโทรว์ผ่านทาง COM อินเทอร์เฟซ, Windows Media Player control หรือทาง Media Control Interfaces (MCI) ดังแสดงให้เห็นในรูป



รูปที่ 2-1 การติดต่อกับไคลเอนต์

เนื่องจากสถาปัตยกรรมของไคลเอนต์นั้นมีความยืดหยุ่น ประกอบไปด้วยหน่วยย่อยหลายหน่วยรวมกัน ฟิวเจอร์ กราฟจะมีความสามารถมากในการนำไปใช้งานและสามารถนำไปประยุกต์ใช้ได้ ยกตัวอย่างเช่น ฟิวเจอร์ กราฟที่ช่วยในการดึงภาพจากวิดีโอ

2.2 สถาปัตยกรรมของไคลเอนต์ฟิวเจอร์

ในการใช้ Filter Graph Manager กับแอปพลิเคชันเราไม่จำเป็นต้องรู้รายละเอียดเกี่ยวกับฟิวเจอร์ กราฟที่นำมาใช้มากนัก แต่อย่างไรก็ตามจะเป็นผลดีถ้าเราเข้าใจหลักสำคัญที่เป็นพื้นฐานของ Filter Graph Manager หากว่าเราต้องการที่จะสร้างฟิวเจอร์ กราฟขึ้นมาใช้เองแทนที่จะปล่อยให้ Filter Graph Manager จัดการให้

ฟิวเจอร์ กราฟประกอบไปด้วยกลุ่มของฟิวเจอร์ที่แตกต่างกัน โดยฟิวเจอร์ส่วนใหญ่สามารถจำแนกได้เป็น 3 ประเภทดังนี้

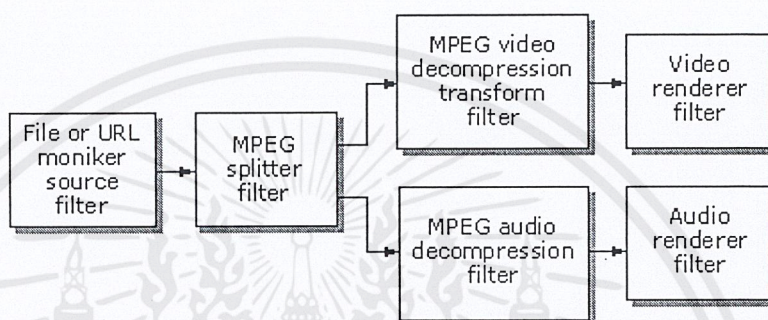
- Source filter ซึ่งรับข้อมูลจากแหล่งที่มา (source) เช่น ไฟล์ที่อยู่ในดิสก์ และนำเข้าสู่ฟิวเจอร์ กราฟ
- Transform filter มีหน้าที่ประมวลผลข้อมูลและส่งต่อไป
- Rendering filter มีหน้าที่เรนเดอร์ (render) ข้อมูลผ่านทางอุปกรณ์ฮาร์ดแวร์ หรือที่ซึ่งรับข้อมูลเข้ามา (อย่างเช่น จากหน่วยความจำ หรือจากไฟล์บนดิสก์)

นอกเหนือไปจาก 3 ชนิดนี้แล้วยังมี ฟิวเจอร์ชนิดอื่นๆอีก เช่น Effect filter ซึ่งจะเพิ่มเอฟเฟกต์ (effect) โดยปราศจากการเปลี่ยนแปลงข้อมูล และ Parser filter ซึ่งจะเข้าใจในรูปแบบของข้อมูลที่รับเข้ามาและรู้ว่าจะอ่านข้อมูลที่ใดบ้างจึงจะถูกต้อง ทั้งยังสร้าง Time stamp และทำการค้นหา

ยกตัวอย่างเช่น ฟิวเจอร์กราฟที่ใช้เพื่อเล่นไฟล์ MPEG วิดีโอ จะใช้ฟิวเจอร์ดังต่อไปนี้

- Source filter เพื่อที่จะอ่านข้อมูลจากดิสก์
- MPEG filter เพื่อจะแบ่งข้อมูลเป็นส่วนๆ แยก MPEG ออดิโอ ออกจาก MPEG วิดีโอ
- Transforms filter เพื่อขยายข้อมูลวิดีโอที่ถูกบีบอัดไว้
- Video renderer filter เพื่อเรนเดอร์ข้อมูลวิดีโอบนฉาก
- Audio renderer filter เพื่อส่งเสียงไปที่ซาวนด์การ์ด

ฟิลเตอร์กราฟแสดงให้เห็นดังภาพต่อไปนี้



รูปที่ 2-2 ฟิลเตอร์ กราฟเพื่อเล่นไฟล์ MPEG

มันเป็นไปได้ที่ตัวฟิลเตอร์จะทำหน้าที่หลายอย่าง อย่างเช่น ฟิลเตอร์อาจจะทำหน้าที่เรนเดอร์เสียง ขณะเดียวกัน ก็ยังทำหน้าที่เป็น transform filter ด้วยโดยการส่งผ่านข้อมูลวิดีโอต่อไป แต่โดยทั่วไปแล้วฟิลเตอร์จะเหมาะกับหน้าที่หนึ่งใน 3 ประเภทที่กล่าวมา

ฟิลเตอร์ กราฟมีการส่งกระแสข้อมูลมัลติมีเดียผ่านฟิลเตอร์ ในกระแสข้อมูลฟิลเตอร์หนึ่งส่งผ่านข้อมูลไปสู่ฟิลเตอร์หนึ่ง upstream filter จะบอกถึงฟิลเตอร์ตัวที่ส่งข้อมูลมาให้ ส่วน downstream filter จะบอกถึงฟิลเตอร์ตัวถัดไปในสายของข้อมูล ความแตกต่างนี้สำคัญ เนื่องจากสื่อข้อมูลจะส่งผ่านแบบ downstream ในขณะที่ข้อมูลชนิดอื่นส่งแบบ upstream ได้

เพื่อที่จะทำให้ฟิลเตอร์ กราฟทำงานได้ ฟิลเตอร์ต้องเชื่อมต่อกันเป็นลำดับ และกระแสข้อมูลต้องเริ่มต้นและสิ้นสุดอย่างเป็นแบบแผน filter graph manager จะทำการเชื่อมต่อฟิลเตอร์และควบคุมสื่อข้อมูล และยังสามารถในการค้นหาฟิลเตอร์เพื่อเรนเดอร์สื่อข้อมูลและสร้างฟิลเตอร์ กราฟ ซึ่งฟิลเตอร์กราฟสามารถกำหนดล่วงหน้าได้ในกรณีที่ filter graph manager ไม่ต้องการที่จะค้นหาฟิลเตอร์อื่น

เมื่อทำการค้นหาฟิลเตอร์ที่ใช้แสดงผล filter graph manager จะใช้ filter mapper โดยที่ตอนแรก จะทำการอ่านรีจิสทรี (registry) และกำหนดชนิดของฟิลเตอร์ที่มีอยู่ filter graph manager จะพยายามเชื่อมฟิลเตอร์ที่รับชนิดข้อมูลนั้นจนกระทั่งมาถึง rendering filter ค่าคุณสมบัติ (property) ของแต่ละฟิลเตอร์จะถูกกำหนดไว้ ซึ่งจะบอกถึงความสามารถของการจัดการกับข้อมูลของฟิลเตอร์ โดยฟิลเตอร์ที่มีค่าคุณสมบัติสูงที่สุดจะถูกนำมาใช้ก่อน

นอกจากการควบคุมสื่อข้อมูลให้ เล่น หยุดค้างไว้ หรือหยุด แล้วยังสามารถให้เล่นในช่วงเวลา

เฉพาะ หรือค้นหาจุดที่ต้องการในสื่อข้อมูล Filter Graph Manager ขอมให้แอปพลิเคชัน หรือ Microsoft เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ActiveX Control ทำสิ่งที่กล่าวมาแล้วได้ โดยเรียกเมธอด (method) ที่เหมาะสมตามที่ฟิลเตอร์ร้องขอมา และยังยอมให้ฟิลเตอร์แจ้งสิ่งที่เกิดขึ้น ที่แอปพลิเคชันสามารถเรียกดูได้ เช่น เรียกดูสถานะของฟิลเตอร์ที่นำมาต่อไว้

2.3 การใช้โคเรกโซว์

มี 3 วิธีในการใช้โคเรกโซว์

- ผ่านทาง Windows Media Player Control ซึ่งจะมีอินเทอร์เฟซที่ง่ายต่อการใช้งานเพื่อจัดการกับมัลติมีเดีย โดยใช้คุณสมบัติของคอนโทรล (control), เมธอด และอีเวนต์ (event) คอนโทรลจะจัดการกับส่วนที่เรนเดอร์ภาพและเสียง ช่วยให้การเขียนโปรแกรมไม่ยุ่งยากและง่ายต่อการเพิ่มเติมมัลติมีเดียไปในเว็บเพจ และแอปพลิเคชันอื่นๆ
- ด้วยฟิลเตอร์ที่มีมาให้โคเรกโซว์มีชุดฟิลเตอร์มาตรฐานมาพร้อมกับชุดพัฒนา โคเรกโซว์ SDK
- สร้างฟิลเตอร์ขึ้นเอง ชุดพัฒนา SDK ช่วยให้สามารถสร้างฟิลเตอร์ขึ้นมาใช้งานเองได้โดยใช้ไลบรารีคลาสของโคเรกโซว์ เบสคลาสในไลบรารีที่เขียนด้วยภาษาซีหรือซีพลัสพลัส จะต้องใช้ COM อินเทอร์เฟซกับฟิลเตอร์และกำหนดโครงสร้างพื้นฐานของฟิลเตอร์ เราสามารถสร้างฟิลเตอร์ของโคเรกโซว์ด้วยภาษาใดๆ ที่สามารถสร้างออบเจกต์จาก COM ได้

เราต้องเรียนรู้เกี่ยวกับเทคโนโลยีที่สนับสนุนงานของเรา เช่น เราต้องเข้าใจ การเขียนโปรแกรมแบบ COM เมื่อใช้ภาษาซีหรือซีพลัสพลัส เพื่อคอนโทรลโคเรกโซว์ Playback หรือสร้างฟิลเตอร์แต่เราไม่จำเป็นต้องเข้าใจการเขียนโปรแกรมแบบ COM เมื่อใช้ ActiveMovie คอนโทรล

2.4 การใช้โคเรกโซว์กับวิซวลเบสิก

เนื้อหาต่อไปนี้จะบอกว่าจะเขียนโคเรกโซว์แอปพลิเคชันโดยใช้วิซวลเบสิกได้อย่างไร

2.4.1 การสร้างฟิลเตอร์โดยใช้วิซวลเบสิก

จะบอกถึงว่าเราจะใช้วิซวลเบสิกจัดการกับส่วนประกอบของฟิลเตอร์ กราฟอย่างไร โดยควรมีความคุ้นเคยกับการเขียนโปรแกรมแอปพลิเคชันบนวินโดวส์ การเขียนโปรแกรมแบบมัลติมีเดียบนวินโดวส์ และลักษณะของการเขียนโปรแกรมด้วยวิซวลเบสิก

2.4.1.1 การสร้างฟิลเตอร์ กราฟ

มี 3 วิธีที่จะเรียกใช้ Quartz.dll เพื่อสร้างฟิลเตอร์ กราฟ แต่ละวิธีจะมีการควบคุมฟิลเตอร์ที่แตกต่างกัน ซึ่งก็คือการสร้างฟิลเตอร์ กราฟทั้งหมดโดยอัตโนมัติเพื่อระบุรายละเอียดแก่ทุกๆ ฟิลเตอร์และการเชื่อมต่อของ pin โดย 3 วิธีที่จะกล่าวถึงมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แบบอัตโนมัติ : สร้างฟิลเตอร์ กราฟโดยสมบูรณ์จากสื่อมัลติมีเดีย หรือฟิลเตอร์ กราฟไฟล์ที่เก็บไว้
- แบบกึ่งอัตโนมัติ : สร้างฟิลเตอร์ กราฟใหม่ (ว่าง) ขึ้นมา แล้วเพิ่มฟิลเตอร์เข้าไปจากนั้นจะเกิดการสร้างฟิลเตอร์ และเชื่อมต่อกัน โดยอัตโนมัติเพื่อทำงานตามหน้าที่ของ pin
- แบบ Manual : สร้างฟิลเตอร์ กราฟใหม่ (ว่าง) ขึ้นมา แล้วเพิ่มแต่ละฟิลเตอร์ไปยังกราฟ และใส่ส่วนที่เชื่อมต่อกันระหว่าง pin

2.4.1.2 การสร้างฟิลเตอร์ กราฟแบบสมบูรณ์

โค้ดต่อไปนี้เป็นส่วนที่แสดงว่าจะสร้างฟิลเตอร์ กราฟแบบสมบูรณ์ได้อย่างไรโดยอ้างอิงจากข้อมูลมัลติมีเดียที่รับมาหรือฟิลเตอร์ กราฟที่เก็บไว้ หลังจากสร้างออบเจกต์ FilgraphManager ซึ่งจะ “empty” ในตอนเริ่มต้น จากนั้นแอปพลิเคชันเรียกเมธอด FilgraphManager.RenderFile เพื่อสร้างกราฟที่สมบูรณ์

```
' Start by creating a new, empty filter graph.
Dim g_objMC as IMediaControl ' from the General Declarations section ...
Set g_objMC = New FilgraphManager ' create the new filter graph
' Use the common File Open dialog to let the user select the input file.
CommonDialog1.ShowOpen ' user selects a source or filter graph
' call IMediaControl.RenderFile to add all filters and connect all pins.
g_objMC.RenderFile CommonDialog1.filename ' Generates the complete graph
```

2.4.1.3 รูปการใช้ออบเจกต์ของโคเรกโซว์ คุณสมบัติและเมธอด

Task	DirectShow properties and methods
สร้าง empty, new ฟิลเตอร์ กราฟ	กำหนด objMediaControl = New FilgraphManager
สร้างแบบฟิลเตอร์ กราฟแบบ สมบูรณ์เฉพาะไฟล์นั้นๆ	เรียกใช้เมธอด FilgraphManager.RenderFile
เพิ่ม Source filter	เรียกใช้เมธอด FilgraphManager.AddSourceFilter
เพิ่ม Rederer หรือ Transform filter	เรียกเอาค่าของออบเจกต์ IRegFilterInfo โดยใช้คุณสมบัติ FilgraphManager.RegFilterCollection; เรียก ใช้เมธอด IRegFilter.Filter
ดูรายการ pin ของออบเจกต์ฟิลเตอร์	เรียกเอาค่าของออบเจกต์ IPinInfo โดยใช้คุณสมบัติ IFilterInfo.Pins
เชื่อมต่อ pin	เรียกใช้เมธอด IPinInfo.Connect
ทำการเชื่อมต่อจากทุก pin ไปสู่ Renderer filter	เรียกใช้เมธอด IPinInfo.Render

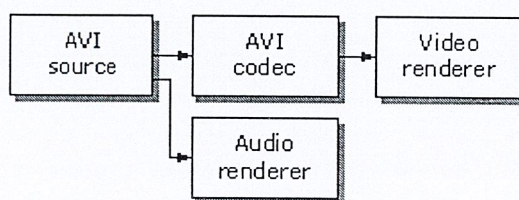
ตารางที่ 2-1 แสดงออบเจกต์ คุณสมบัติและเมธอดในไคเรกโซว์

2.4.2 การควบคุมฟิลเตอร์ กราฟโดยใช้วิซวลเบสิก

จะบอกถึงว่าจะใช้เมธอด, อีเวนต์และคุณสมบัติ ที่อยู่ใน ไคเรกโซว์ Dynamic-Link Library, Quartz.dll เพื่อที่จะเรนเดอร์ข้อมูล Time-stamped ของวิดีโอด้วยวิซวลเบสิก

2.4.2.1 ไคเรกโซว์ฟิลเตอร์และฟิลเตอร์ กราฟ

เมื่อมีลิตมีเดียถูกแสดงในแอปพลิเคชันโดยใช้ Quartz.dll นั่นคือ แอปพลิเคชันกำลังใช้กลุ่มของออบเจกต์ที่เรียกว่าฟิลเตอร์ หรือบางครั้งเรียกว่า ฟิลเตอร์ กราฟ ไคอะแกรมต่อไปนี้จะแสดงรูปฟิลเตอร์ที่สามารถเรนเดอร์ไฟล์ AVI (Audio-Video Interleaved)



รูปที่ 2-3 ฟิลเตอร์ของไฟล์ AVI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโคเดแกรม AVI source filter จะอ่านไฟล์จากดิสก์ จากนั้น AVI Decompressor filter (codec) ทำการขยายข้อมูลวิดีโอที่ถูกบีบอัดขณะที่ส่งมาจาก Source filter ต่อมา Codec filter ส่งผ่านข้อมูลไปยัง Video renderer ซึ่งจะส่งข้อมูลไปยังอุปกรณ์ในรูปแบบที่อุปกรณ์นั้นเข้าใจ AVI Source filter ส่งข้อมูลเสียงโดยตรงไปยัง Audio renderer ซึ่งจะส่งข้อมูลไปยังอุปกรณ์ที่ให้เสียง

นอกจากฟิลเตอร์แล้ว ไดรกโครยังสนับสนุนออบเจกต์อัตโนมัติที่เรียกว่า Filter Graph Manager ออบเจกต์นี้จะรู้เกี่ยวกับฟิลเตอร์ที่มีอยู่และเข้าใจชนิดของฟิลเตอร์ที่ต้องการ เพื่อจะแสดงรูปแบบของไฟล์ชนิดต่างๆ Filter Graph Manager จะแสดงเมธอด, อีเวนต์และคุณสมบัติที่สามารถใช้ได้กับฟิลเตอร์ในฟิลเตอร์ กราฟนั้น Filter Graph Manager ยังมีเมธอด, อีเวนต์และคุณสมบัติของตัวเองอีกด้วย ทั้งหมดนี้สามารถถูกแสดงให้เห็นโดยใช้อินเทอร์เฟซ ซึ่งเป็นกลุ่มของเมธอด, อีเวนต์และคุณสมบัติที่เกี่ยวข้อง

ตารางต่อไปนี้บอกถึงอินเทอร์เฟซที่มีอยู่ใน Quartz.dll เพื่อใช้กับวิซวลเบสิก แอปพลิเคชัน (Visual Basic Application) และแจกแจงถึงรายละเอียดของแต่ละอินเทอร์เฟซ

Interface	Description
IAMcollection	ติดต่อกับ pin และฟิลเตอร์อื่นๆ
IBasicAudio	ควบคุมและเรียกใช้ค่าเสียงที่ตั้งไว้
IBasicVideo2	คอนโทรลส่วนแสดงภาพทัวๆไป
IFilterInfo	เรียกดูข้อมูลเกี่ยวกับฟิลเตอร์และที่เกี่ยวข้องกับออบเจกต์ pin ในฟิลเตอร์
FilterGraphManager	สร้างฟิลเตอร์ กราฟและควบคุมการไหลของข้อมูล (เล่น, หยุดค้าง, หยุด)
IMediaEvent	อนุญาตให้เปลี่ยนแปลงส่วนที่จัดการกับอีเวนต์ เช่น repainting, user termination, completion
IMediaPosition	ควบคุมและเรียกใช้ เวลาเริ่มต้น เวลาหยุด อัตราเร็ว และ ตำแหน่งปัจจุบัน
IMediaTypeInfo	เรียกดูชนิดของข้อมูลและข้อมูลย่อย
IPinInfo	ติดต่อกับข้อมูลของ pin เช่น ทิศทางและชนิดข้อมูล และควบคุมการเชื่อมต่อของ pin การตัดขาด การแสดงผล
IRegFilterInfo	เก็บรายละเอียดของฟิลเตอร์(Transform and Render)
IVideoWindow	ควบคุมหน้าต่างเรนเดอร์ภาพ

ตารางที่ 2-2 แสดงอินเทอร์เฟซของ Quartz.dll

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.2 การควบคุมฟิลเตอร์ กราฟในวิซวลเบสิก

2.4.2.2.1 การเตรียมใช้ออบเจกต์ของไคเรกโซว์

วิซวลเบสิกเริ่มต้นทุกออบเจกต์โดยใช้ FilgraphManager ซึ่งสนับสนุนอินเทอร์เฟซต่อไปนี้

- IBasicAudio
- IBasicVideo2
- IMediaEvent
- IMediaPosition
- IVideoWindow

แต่ละอินเทอร์เฟซถูกเรียกใช้โดยออบเจกต์ของวิซวลเบสิก ที่สามารถกำหนดได้ว่าจะเป็นอินเทอร์เฟซชนิดไหน เราสามารถประกาศออบเจกต์ให้เป็นตัวแปรแบบโกลบอล (global) ดังตัวอย่างต่อไปนี้

```
Dim g_objVideoWindow As IVideoWindow
Dim g_objMediaControl As FilgraphManager
Dim g_objMediaPosition As IMediaPosition
Dim g_objBasicAudio As IBasicAudio
Dim g_objBasicVideo As IBasicVideo2
```

Initialize ออบเจกต์โดยใช้ FilgraphManager ดังแสดงดังตัวอย่าง

```
Set g_objMediaControl = New FilgraphManager
g_objMediaControl.RenderFile (g_strFileName) ' name of input file ...
Set g_objBasicAudio = g_objMediaControl
Set g_objVideoWindow = g_objMediaControl
Set g_objMediaEvent = g_objMediaControl
Set g_objMediaPosition = g_objMediaControl
```

2.4.2.2.2 การ Instantiating ฟิลเตอร์ กราฟ

เราสามารถใส่ filter graph manager เพื่อแสดงฟิลเตอร์ กราฟที่มีอยู่แล้วโดยระบุถึงไฟล์ที่อยู่ในกราฟให้เป็นพารามิเตอร์ ใช้เมธอด FiltergraphManager.RenderFile

เนื่องจากฟิลเตอร์ที่อยู่ในฟิลเตอร์ กราฟขึ้นอยู่กับชนิดของไฟล์ที่จะถูกนำมาแสดง ดังนั้นห้าม instantiate ฟิลเตอร์ กราฟจนกระทั่งเลือกไฟล์แล้ว Filter Graph Manager จะสร้าง instantiate เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิชาลเบติกใช้คีวีเวิร์ด NEW เพื่อสร้างออบเจ็กต์โดยอัตโนมัติ ออบเจ็กต์ฟิลเตอร์ กราฟถูกสร้างเมื่อเมธอด RenderFile ถูกนำมาใช้ ดังแสดงตามตัวอย่าง

```
'Instantiate a filter graph for the requested
'file format.
Set g_objMediaControl = New FileGraphManager
g_objMediaControl.RenderFile (g_strFileName)
```

2.4.2.2.3 การเรนเดอร์วิดีโอ

ออบเจ็กต์ FilterGraphManager สนับสนุน 3 เมธอดที่แอปพลิเคชันสามารถเรียกใช้เพื่อควบคุม video stream: Run, Pause และ Stop หลังจากออบเจ็กต์ฟิลเตอร์ กราฟถูก instantiate แอปพลิเคชันสามารถเรียกใช้เมธอดเหล่านี้

โค้ดต่อไปนี้เป็นตัวอย่างการเรียกเมธอดในการตอบสนองเมื่อผู้ใช้งานปุ่ม

```
Private Sub Toolbar1_ButtonClick(ByVal Button As Button)
```

```
' handle buttons on the toolbar
```

```
' if the objects aren't defined, avoid errors
```

```
If g_objMediaControl Is Nothing Then
```

```
Exit Sub
```

```
End If
```

```
If Button.Index = 1 Then 'PLAY
```

```
'Invoke the MediaControl Run() method
```

```
'and play the video through the predefined
```

```
'filter graph.
```

```
g_objMediaControl.Run
```

```
g_fVideoRun = True
```

```
ElseIf Button.Index = 2 Then 'PAUSE
```

```
'Invoke the MediaControl Pause() method
```

```
'and pause the video that is being
```

```
'displayed through the predefined
```

```
'filter graph.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
g_objMediaControl.Pause
```

```
g_fVideoRun = False
```

```
ElseIf Button.Index = 3 Then 'STOP
```

```
'Invoke the MediaControl Stop() method
```

```
'and stop the video that is being
```

```
'displayed through the predefined
```

```
'filter graph.
```

```
g_objMediaControl.Stop
```

```
g_fVideoRun = False
```

```
' reset to the beginning of the video
```

```
g_objMediaPosition.CurrentPosition = 0
```

```
txtElapsed.Text = "0.0"
```

2.4.2.2.4 การควบคุมเสียง

อินเทอร์เฟซ IBasicAudio สนับสนุน 2 คุณสมบัติคือ Volume และ Balance คุณสมบัติ Volume คือเรียกดูหรือตั้งค่าความดัง คุณสมบัติ Balance คือเรียกดูหรือตั้งค่า Balance เสียง ตัวอย่างต่อไปนี้จะปรับค่าความดังโดยตั้งค่าคุณสมบัติ Volume

```
Private Sub slVolume_Change()
```

```
'Set the volume on the slider
```

```
If Not g_objMediaControl Is Nothing Then
```

```
'if g_objMediaControl has been assigned
```

```
g_objBasicAudio.Volume = slVolume.Value
```

```
End If
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.2.5 การปรับเปลี่ยนสเกลวิดีโอเอาต์พุต

อินเทอร์เฟซ IVideoWindow สนับสนุนเมธอดและคุณสมบัติที่ปรับ size, state, owner, palette, visibility ได้หลากหลายและอื่นๆสำหรับหน้าต่างแสดงภาพ ถ้าไม่ได้ตั้งค่าขนาดของหน้าต่างจะแสดงผลลัพท์ในหน้าต่างขนาดปกติ (ซึ่งจะปรากฏที่มุมซ้ายบนของจอภาพ) โดยปราศจากการเรียกใช้เมธอดหรือคุณสมบัติใดๆเหล่านี้

เราสามารถเปลี่ยนรูปแบบหน้าต่างโดยการย้าย caption, border และ dialog box frame โดยต้องตั้งค่าคุณสมบัติ IVideoWindow.WindowStyle เป็น 0x06000000 ซึ่งจะไปทำโอเปอเรชัน OR กับค่า WS_DLGMFRAME (0x04000000) และ WS_VSCROLL (0x02000000)

เพื่อที่จะย้ายหน้าต่างบนฟอร์ม ระบุตำแหน่งใหม่โดยตั้งค่าคุณสมบัติ IVideoWindow.Top และ IVideoWindow.Left ของ g_objVideoWindow คุณสมบัติ Top และ Left ถูกตั้งเพื่อให้มุมซ้ายบนเกิดขึ้นที่ว่างบนฟอร์ม คุณสมบัติ ScaleMode สำหรับฟอร์มถูกตั้งเป็น 3 ซึ่งระบุว่ามีหน่วยเป็นพิกเซล (pixel) ซึ่งจะทำให้คุณสมบัติของฟอร์มและคุณสมบัติของออบเจกต์ไคเรกโซว์ ถูกใช้โดยปราศจากการเปลี่ยนแปลง คุณสมบัติของออบเจกต์ไคเรกโซว์ ถูกแสดงในหน่วยพิกเซล หน่วยเดิมของฟอร์มคือ twip

กำหนดขนาดที่ต้องการของสี่เหลี่ยมแสดงภาพโดยการเรียกดูความกว้างและความสูงของวิดีโอ ค่าเหล่านี้จะตรงกับคุณสมบัติ IBasicVideo2.VideoWidth และ IBasicVideo2.VideoHeight

นอกจากนี้การตั้งค่าคุณสมบัติ Top และ Left จำเป็นที่จะต้องระบุฟอร์มของแอปพลิเคชันให้เป็น parent window ใหม่โดยการส่งให้ส่วนควบคุมหน้าต่างของฟอร์ม hWnd ให้แก่คุณสมบัติ IVideoWindow.Owner ถ้าการควบคุมไม่ถูกส่ง หน้าต่างจะปรากฏเป็นพื้นหลัง และไม่อยู่บนฟอร์ม

แสดงตัวอย่างดังต่อไปนี้

```
Set g_objVideoWindow = g_objMediaControl
g_objVideoWindow.WindowStyle = CLng(&H6000000) ' WS_DLGMFRAME | WS_VSCROLL
g_objVideoWindow.Left = CLng(Shape1.Left) ' shape is a placeholder on the form
g_objVideoWindow.Top = CLng(Shape1.Top)
Shape1.Width = g_objVideoWindow.Width ' resize the shape given the input video
Shape1.Height = g_objVideoWindow.Height
g_objVideoWindow.Owner = frmMain.hWnd ' set the form as the parent
```

ต่อไปจะแสดงการตั้งค่าคุณสมบัติ ScaleMode

```
' ...
frmMain.ScaleMode = 3 ' pixels
' ...
```

อย่าพยายามที่จะตั้งสเกลหน้าต่าง โดยตั้งค่าคุณสมบัติ `IVideoWindow.Width` และ `IVideoWindow.Height` แก่ออบเจกต์ `IVideoWindow` เนื่องจากต้องตระหนักถึงข้อจำกัดของขนาดด้วย

2.4.2.2.6 การติดตามสถานะ

ออบเจกต์ `IMediaPosition` มีคุณสมบัติที่เราสามารถใช้เพื่อเรียกดูหรือตั้งค่า ตำแหน่งปัจจุบัน, จุดหยุด, ระยะเวลาและ อัตราเร็ว โค้ดต่อไปนี้จะแสดงให้ดูว่าจะทำได้อย่างไร

```
Set g_objMediaPosition = g_objMediaControl
g_dblRunLength = g_objMediaPosition.Duration
txtDuration.Text = CStr(g_dblRunLength) ' display the duration
g_dblStartPosition = 0.0
txtStart.Text = CDbl(g_dblStartPosition) ' display the start time
g_dblRate = g_objMediaPosition.Rate
txtRate.Text = CStr(g_dblRate)
```

2.4.2.2.7 การรับและตั้งค่าตำแหน่งเริ่มต้น

ใช้คุณสมบัติ `IMediaPosition.CurrentPosition` เพื่อให้ผู้ใช้ปรับจุดที่ภาพจะเริ่มเรนเดอร์ โค้ดตัวอย่างต่อไปนี้จะแสดงการเปลี่ยนตำแหน่งเมื่อผู้ใช้กดคีย์

```
Private Sub txtStart_KeyDown(KeyCode As Integer, Shift As Integer)
' handle user input to change the start position
If KeyCode = vbKeyReturn Then
If g_objMediaPosition Is Nothing Then
Exit Sub
ElseIf CDbl(txtStart.Text) > g_dblRunLength Then
MsgBox "Specified position invalid: re-enter new position."
ElseIf CDbl(txtStart.Text) < 0 Then
MsgBox "Specified position invalid: re-enter new position."
ElseIf CDbl(txtStart.Text) <> "" Then
g_dblStartPosition = CDbl(txtStart.Text)
g_objMediaPosition.CurrentPosition = g_dblStartPosition
End If
End If
End Sub
```

2.4.2.2.8 การรับและตั้งค่าอัตราเร็ว

ใช้คุณสมบัติของ IMediaPosition.Rate เพื่อให้ผู้ใช้สามารถปรับอัตราเร็วที่ภาพถูกเรนเดอร์ อัตราเร็วนี้เป็นอัตราส่วนที่เกี่ยวกับความเร็วเมื่อเล่นวิดีโอปกติ ตัวอย่าง อัตราเร็ว 0.5 ทำให้ภาพถูกเรนเดอร์ที่ความเร็วครึ่งหนึ่งของความเร็วปกติ และที่อัตราเร็ว 2.0 ทำให้ภาพถูกเรนเดอร์เป็น 2 เท่าของความเร็วปกติ

ต่างจากคุณสมบัติ CurrentPosition ซึ่งสามารถตั้งค่าในขณะที่ภาพถูกเรนเดอร์ คุณสมบัติ Rate ต้องตั้งค่าก่อนที่จะเรนเดอร์ แทร็กเสียงอาจใช้ไม่ได้สำหรับบางวิดีโอเมื่ออัตราเร็วน้อยกว่า 1.0 โต้ตอบไปนี้แสดงตัวอย่างการเปลี่ยนอัตราเร็วเมื่อผู้ใช้กดคีย์

```
Private Sub txtRate_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
' DirectShow VB sample
```

```
' handle user updates to the Rate value
```

```
If KeyCode = vbKeyReturn Then
```

```
    If g_objMediaPosition Is Nothing Then
```

```
        Exit Sub
```

```
    ElseIf CDbl(txtRate.Text) < 0# Then
```

```
        MsgBox "Negative values invalid: re-enter value between 0 and 2.0"
```

```
    ElseIf CStr(txtRate.Text) <> "" Then
```

```
        g_dblRate = CDbl(txtRate.Text)
```

```
        g_objMediaPosition.Rate = g_dblRate
```

```
    End If
```

```
End If
```

```
End Sub
```

2.4.2.2.9 Cleaning up

แต่ละครั้งที่แอปพลิเคชันใช้ คำสั่ง Set เพื่อสร้างอินสแตนซ์ใหม่ของออบเจ็กต์ใดเรกโซว์ จะต้องใช้คำสั่ง Set อีกเพื่อที่จะลบออบเจ็กต์และสิ่งที่เกี่ยวข้องนั้นจากหน่วยความจำก่อนจะจบโปรแกรมดังกล่าว

```
Set g_objBasicAudio = Nothing
```

2.4.2 ออบเจ็กต์ในวิซวลเบสิก

ไคเรกโซว์มีการใช้งานในวิซวลเบสิกโดยใช้ออบเจ็กต์เพื่อสร้างและควบคุมฟิลเตอร์กราฟ

Object	Description
FilgraphManager	สร้างและควบคุมฟิลเตอร์ กราฟ
IAMCollection	แสดงชุดของ pin ฟิลเตอร์ และชนิดของสื่อ
IBasicAudio	สนับสนุนส่วนประกอบด้านเสียงของฟิลเตอร์ กราฟ
IBasicVideo2	สนับสนุนส่วนประกอบด้านภาพวิดีโอของฟิลเตอร์ กราฟ
IFilterInfo	จัดการข้อมูลเกี่ยวกับฟิลเตอร์
IMediaEvent	บรรจุเมธอดสำหรับเรียกดูสิ่งที่เกิดขึ้น
IMediaPosition	บรรจุเมธอดสำหรับค้นหาฟิลเตอร์กราฟและตั้งค่า Playback
IMediaTypeInfo	จัดการกับข้อมูลเกี่ยวกับชนิดของสื่อ
IRegFilterInfo	จัดการกับข้อมูลเกี่ยวกับ pin ของฟิลเตอร์
IVideoWindow	จัดการกับหน้าต่างของ Video renderer

ตารางที่ 2-3 แสดงออบเจกต์ในวิซวลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

DES (DirectShow Editing Services)

3.1 DirectShow Editing Services

บทนี้จะกล่าวถึง DirectShow Editing Services (DES) ซึ่งเป็นเอพีไอสนับสนุนการตัดต่อวิดีโอ

3.1.1 DirectShow Editing Services เบื้องต้น

DES เป็นเอพีไอที่ทำให้การตัดต่อวิดีโอเป็นกลายเป็นเรื่องง่าย DES ถูกสร้างบนสถาปัตยกรรมหลักของไคลเรกโซว์ มันเป็นภาพรวมความซับซ้อนของไคลเรกโซว์ และมีชุดอินเทอร์เฟซที่ออกแบบมาโดยเฉพาะสำหรับงานตัดต่อวิดีโอ ในฐานะนักพัฒนาแอปพลิเคชัน เราได้ประโยชน์จากไคลเรกโซว์ในด้านของโครงสร้างของงานมากกว่าการสร้างแอปพลิเคชันตัดต่อวิดีโอ

แกนหลักของไคลเรกโซว์เป็นสถาปัตยกรรมที่มีศักยภาพมากสำหรับจัดการกับกระแสสื่อข้อมูล แอปพลิเคชันสามารถใช้เพื่อเล่น ไฟล์มัลติมีเดียที่มีอยู่หลากหลายรูปแบบ โดยปราศจากการที่นักพัฒนาต้องกังวลกับการบีบอัดข้อมูลและรายละเอียดอื่นๆที่น่ารำคาญ ก่อนที่จะพูดถึง DES ใดๆก็ตามไคลเรกโซว์นั้นยังขาดความยืดหยุ่นที่จำเป็นสำหรับการตัดต่อแบบ nonlinear

สมมุติว่าเราต้องการที่จะสร้างวิดีโอที่ประกอบด้วย 4 วินาทีจาก Source A ตามด้วย 10 วินาทีจาก Source B และจบด้วย 5 วินาทีจาก Source C เราสามารถทำได้โดยไม่ยากโดยใช้เอพีไอหลักของไคลเรกโซว์

แต่ถ้าเราคิดว่า Source C น่าจะมาก่อน Source B ไม่ใช่ตามหลัง ลำดับจะใช้เวลา 8 วินาทีจาก Source A ไม่ใช่ 4 และผลที่ได้ทั้งหมดจำเป็นต้องแยกเสียงออกด้วยหรือไม่ แม้ว่าการเปลี่ยนแปลงเล็กน้อยนี้จะทำได้ยาก แต่สิ่งเล็กน้อยเหล่านี้เมื่อทำการตัดต่อวิดีโอด้วย DES เราสามารถทำได้ด้วยการเรียกใช้เมธอดต่างๆ

นี่คือลักษณะบางอย่างที่ DES นำมาใช้กับไคลเรกโซว์

- แบบจำลองที่เรียกว่า Timeline ซึ่งประกอบด้วยแทร็กภาพ (video track) และเสียง (audio track) หลายแทร็กในเลเยอร์ (layer) เดียวกันทำให้ง่ายต่อการทำงาน
- ความสามารถที่จะแสดงวิดีโอ
- งานอยู่ในรูปแบบของ XML
- สนับสนุนเอฟเฟกต์ทั้งภาพและเสียง รวมทั้ง Transition ระหว่างแทร็กวิดีโอ (เช่น fades และ wipes)
- มีถึง 100 มาตรฐานของการทำ Wipe ซึ่งกำหนดโดย Society of Motion Picture and Television Engineers (SMPTE)
- มีคีย์หลักขึ้นกับ สี, luminance, ค่า RGB หรือ ค่า alpha
- แปลงอัตราเฟรม (frame rate) และ Audio Sampling Rate โดยอัตโนมัติ ทำให้สามารถทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างโดยใช้ Heterogeneous Source

- เปลี่ยนขนาดหรือเก็บภาพ

3.1.2 หลักการพื้นฐานของ DES

3.1.2.1 DES และ เอพีโอหลักของไดเรกโซว์

Timeline ใน DES เป็นภาพรวมรายละเอียดต่างๆของโปรเจกต์ตัดต่อวิดีโอ มันจะระบุถึงวิดีโอที่ใช้ในโปรเจกต์ เวลาเริ่มต้นและหยุดของวิดีโอ ตำแหน่งของเอฟเฟกต์หรือ Transition อย่างไรก็ตาม Timelineไม่สามารถที่จะเรนเดอร์ภาพและเสียงได้ ซึ่งจะเป็นหน้าที่ของ Render engine ซึ่งจะแปลง Timeline เป็นฟิลเตอร์ กราฟ เมื่อฟิลเตอร์ กราฟทำการเรนเดอร์โปรเจกต์ ไม่ว่าจะแสดงโดยตรงหรือเป็นไฟล์ แอปพลิเคชันดำเนินการกับ Timeline มากกว่าที่จะทำกับฟิลเตอร์ กราฟโดยตรง ซึ่งจะยุ่งยากและเกิดความผิดพลาด

3.1.3 แบบจำลอง Timeline

Timeline เป็นออบเจกต์ที่ DES ใช้เพื่อแทนโปรเจกต์การตัดต่อวิดีโอ หัวข้อนี้จะพูดถึงหลักการรายละเอียดของออบเจกต์ต่างๆที่มีอยู่ใน Timeline

3.1.3.1 โครงสร้างของ Timeline

การสร้างวิดีโอเริ่มจากไฟล์ต้นฉบับหลายๆไฟล์เช่น ไฟล์ภาพ ไฟล์เสียง และภาพ ผู้สร้างตัดส่วนของไฟล์เหล่านี้ หลังจากนั้นนำมาจัดเรียงเป็นลำดับ

ในกรณีที่ยากที่สุด คลิปจะถูกจัดเรียงอย่างเป็นลำดับแต่ละคลิปต่อกันไป การจัดเรียงแบบนี้อยู่ในรูปแบบที่เรียกว่า แทร็ก ดังนั้นการสร้างที่ยากที่สุดคือใส่แทร็กภาพและแทร็กเสียงอย่างละแทร็ก การสร้างที่ซับซ้อนกว่านี้ใช้หลายแทร็ก เช่น แทร็กเสียงสามารถผสมเข้าด้วยกันได้ หรือสามารถใส่เอฟเฟกต์เสียง แบบที่แทร็กหนึ่งสลับเข้าออกกับอีกแทร็กหนึ่ง หรือถ้าเป็นวิดีโอ แทร็กหนึ่งสามารถเลื่อนไปยังอีกแทร็กหนึ่งได้โดยการ fade หรือ wipe ซึ่งเป็นตัวอย่างที่เรียกว่า Compositing ผู้สร้างวิดีโอใส่คลิปจัดตามเวลา (คลิปเรียงต่อกัน) และอยู่ในเลเยอร์ (คลิปประกอบด้วยคลิปอื่นๆ) ใน DES เราจะแทนส่วนประกอบโดยใช้แผนภูมิต้นไม้ (tree)

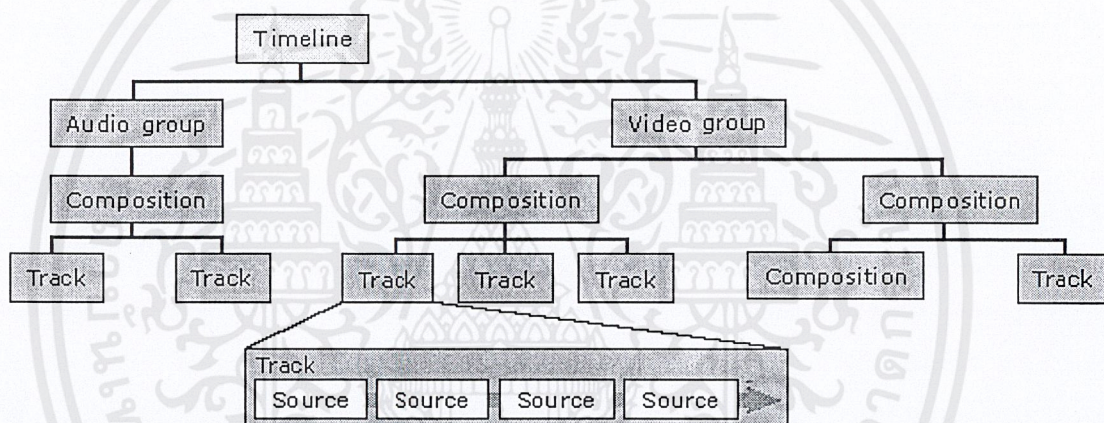
- โหนดปลายแทนสตรีม (stream) ของเสียงหรือภาพ ที่มาจากมีเดียไฟล์ แต่ละโหนดเรียกว่า source
- กลุ่มของ source กับชนิดของมีเดียที่เหมือนกัน (ภาพหรือเสียง) เรียกว่า แทร็ก
- กลุ่มของแทร็ก เรียกว่า Composition ซึ่งจะถูกรเรนเดอร์เสมือนเป็นส่วนประกอบของทุกแทร็กอยู่ข้างในตัวมัน และสามารถเก็บ Composition อื่นๆ ซึ่งสามารถจัดเรียงแทร็กอย่างซับซ้อนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กลุ่มของ Composition และ แทร็ก (ต้องเป็นมีเดียชนิดเดียวกัน) ในระดับ Top-level เรียกว่า กรุป (group)
- กลุ่มของกรุปหรือหลายกรุปรวมเป็น Timeline ซึ่งคือโหนดราก (root node) ในแผนภูมิต้นไม้

Timeline ต้องประกอบด้วยอย่างน้อยหนึ่งกรุป แต่ละกรุปแทนสตรีมในผลลัพธ์สุดท้าย การสร้างจะรวมวิดีโอกรุปและออดิโอกรุป Composition อาจจะมีหรือไม่มีก็ได้ ซึ่งจะมีเพื่อวางโครงสร้างต่อไปอีกถ้าต้องการ

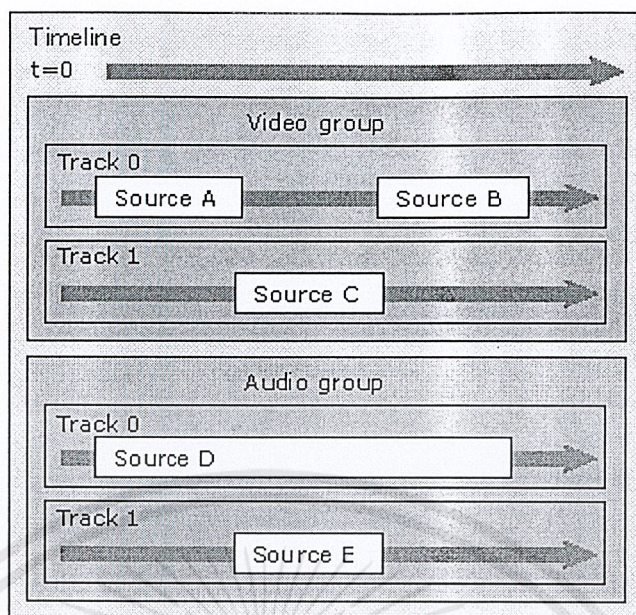
รูปต่อไปนี้จะแสดงโครงสร้างของ Timeline แสดงความสัมพันธ์แบบ child-parent แต่ไม่เป็นไปตามเวลา



รูปที่ 3-1 แสดง โครงสร้างของ Timeline

3.1.3.2 จะเรนเดอร์ Timeline ได้อย่างไร

ไดอะแกรมต้นไม้ในหัวข้อที่แล้วไม่ได้แสดงว่าจะเรนเดอร์แทร็กที่ซ้อนและประกอบกันอยู่ได้อย่างไร สำหรับจุดประสงค์ก็คือ จะช่วยในการพิจารณา Timeline เป็นลำดับเวลา มากกว่าที่จะเป็นโครงสร้างโหนด รูปต่อไปนี้จะแสดงโปรเจกต์ตัวอย่างที่ประกอบด้วย 2 แทร็กภาพ และ 2 แทร็กเสียง ลูกศรด้านบนแทนทิศทางของ Timeline เริ่มต้นที่เวลา 0



รูปที่ 3-2 แสดงการเรนเดอร์ Timeline

อันดับแรกพิจารณาวิดีโอกรุป แทร็ก 1 มีลำดับความสำคัญสูงกว่าแทร็ก 0 เมื่อไรก็ตามที่แทร็ก 1 มีวิดีโออยู่จะซ้อนทับสิ่งที่อยู่ในแทร็ก 0 เมื่อแทร็ก 1 วางแทร็ก 0 ก็จะแสดงให้เห็น และเมื่อวิดีโอถูกเรนเดอร์ จะเริ่มจาก Source A จากนั้นตัดไปยัง Source C ซึ่งจะบังส่วนท้ายของ Source A และตอนเริ่มต้นของ Source B สุดท้ายเมื่อ Source C จบจะตัดไปที่ Source B (ต่อเนื่องไปเรื่อยๆ)

กรุปเสียงมีลักษณะที่แตกต่างไป ทุกแทร็กจะถูกผสมเข้าด้วยกัน เสียงเริ่มจาก Source D จากนั้นเปลี่ยนไปเป็นการผสมกันระหว่าง Source D และ Source E และจบด้วย Source D อีกครั้ง

3.1.4 เวลาที่เกี่ยวข้องใน DES

ในการตัดต่อวิดีโอ เราจะต้องทำงานกับระยะเวลา ดังเช่น

- แต่ละคลิปจะมีระยะเวลา
- คลิป, Transition, เอฟเฟกต์ จะอยู่ในเวลาที่แน่นอน
- วิดีโอมีอัตราเฟรม ซึ่งแสดงในรูป เฟรมต่อวินาที (fps)

DES ประกอบด้วยหลายเมธอดที่เกี่ยวข้องกับเวลาหรืออัตราเฟรม และความหมายของค่าเหล่านี้ขึ้นอยู่กับคำอธิบาย ต่อไปนี้จะอธิบายถึงเครื่องวัดเวลาที่ใช้ใน DES ซึ่งมีดังนี้

3.1.4.1 Time Value

เมื่อมีค่าที่แสดงถึงเวลา มี 3 ความหมายที่สามารถเป็นไปได้

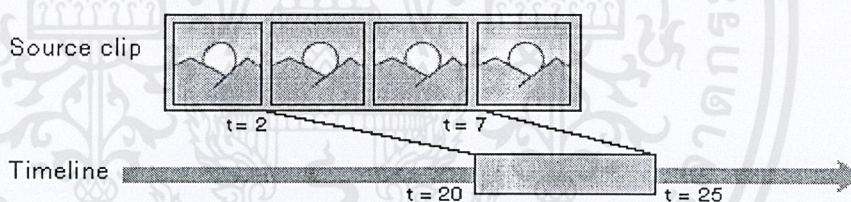
- Timeline time เป็นเวลาที่เกี่ยวข้องกับเวลาเริ่มต้นของ timeline เช่น คลิปอาจจะเริ่มต้น 2 วินาทีใน Timeline, หรือ Transition อาจเกิดขึ้น 15 วินาทีใน Timeline โดยที่ Timeline เป็นตัวกำหนดขึ้น งานที่จะถูกเรนเดอร์ ดังนั้นสามารถพูดได้ว่า Timeline time คือ “project time”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Media Time คือจุดใน Source file ที่เกี่ยวข้องกับตอนเริ่มต้นของไฟล์ ในขณะที่ Playback ปกติ เช่นถ้าไฟล์วิดีโอมี 10 วินาที จุดกึ่งกลางของไฟล์คือ 5 วินาที แสดงเป็น Media time
- Parent time เป็นเวลาที่เกี่ยวเนื่องกับออบเจกต์ใน Timeline เช่น ถ้าออบเจกต์หนึ่งเริ่มต้น 8 วินาทีใน Timeline และมีอีกออบเจกต์ที่เริ่มที่ 10 วินาทีใน Timeline นั่นก็คือ ออบเจกต์ child เริ่มเป็นเวลา 2 วินาทีเกี่ยวเนื่องกับ parent ทุกๆ Virtual track (แทร็กหลายแทร็กที่ประกอบเข้าด้วยกันเปรียบเสมือนแทร็กเดียว) จะเริ่มต้นที่เวลา 0 เกี่ยวเนื่องกับ Timeline ส่วนออบเจกต์ใน Virtual track นั้น Parent time จะเท่ากับ Timeline time

Media time ใช้ได้กับ ออบเจกต์ source แต่ละ source จะมีเวลาเริ่มต้นและเวลาหยุด เช่น ถ้าเรามี วิดีโอ 10 วินาทีและเราต้องการใช้เพียง 5 วินาทีจากตรงกลางของคลิป โดยตัด 2 วินาทีแรกและ 3 วินาทีสุดท้ายออก ถ้าเราต้องการให้ชิ้นงานมี 10 วินาที (สมมุติเป็นอัตรา Playback ปกติ) เราจะต้องระบุเวลาเริ่มต้นและเวลาหยุดต่อไปนี้

- Media เริ่มที่วินาทีที่ 2
- Media หยุดที่วินาทีที่ 7
- Timeline เริ่มที่วินาทีที่ 20
- Timeline หยุดที่วินาทีที่ 25



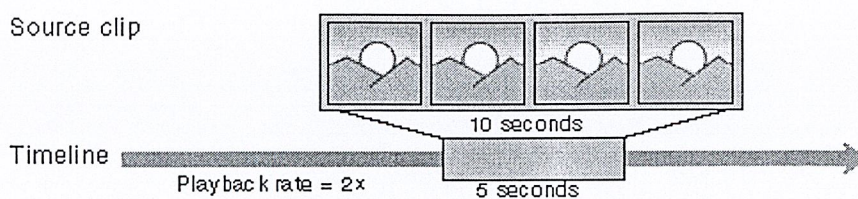
รูปที่ 3-3 แสดงเวลาที่ใช้ในโปรเจกต์

3.1.4.2 Frame Rates

อัตราเฟรมคืออัตราเร็วของกระแสมิติเดียว วัสดุออกมาในรูปแบบต่อวินาที เหมือน time value ความหมายของอัตราเฟรมขึ้นอยู่กับคำอธิบาย

- Output frame rate หมายถึงอัตราเฟรมของโปรเจกต์ที่เรียกว่ากรุป เมื่อเรนเดอร์โปรเจกต์แต่ละกรุปจะเป็นมิติเดียวกันที่มีอัตราเฟรมของตนเอง
- Source frame rate เป็นอัตราเฟรมของไฟล์ต้นฉบับ ซึ่งอาจไม่เท่ากับอัตราเฟรมของกรุป DES จะทำการ Upsample หรือ Downsample ไฟล์ให้โดยอัตโนมัติเท่าที่จำเป็น สำหรับรูปแบบของมิติส่วนใหญ่ DES สามารถทำการกำหนดอัตราเฟรมโดยพิจารณาจากแบบมิติเดียว

Playback rate คืออัตราเร็วของคลิปที่อยู่ในชิ้นงาน เช่น วิดีโอ 10 วินาทีสามารถทำให้เหลือ 5 วินาทีใน Timeline ผลก็คือ ความเร็วของคลิปเพิ่มขึ้น โดยคูณ 2 ดังรูปที่แสดงต่อไปนี้



รูปที่ 3-4 แสดง playback rate

ถ้าเป็น Audio source ระดับเสียงจะเลื่อนไป สูตรต่อไปนี้จะกำหนด Playback rate ของคลิป

- $\text{Playback rate} = (\text{Media Stop} - \text{Media Start}) / (\text{Timeline Stop} - \text{Timeline Start})$

ถ้าไว้ว่าแต่ละอัตราเป็นอิสระต่อกัน:

- สามารถเพิ่มหรือลดความเร็วของคลิปโดยปรับ Media time ซึ่งจะไม่มีผลกระทบต่อผลลัพธ์สุดท้าย
- สามารถเพิ่มหรือลดอัตราเฟรมของผลลัพธ์โดยไม่มีผลต่อความเร็วของการเล่นไฟล์
- สามารถผสม, ภายในกรุปเดียวกันมี Source file ที่มีอัตราเฟรมต่างกัน DES จะ Upsample หรือ Downsample แต่ละคลิปให้ตรงกับอัตราเฟรมของกรุป

เมื่อเรเนเดอร์โปรเจกต์ เวลาจะถูกกะประมาณให้ใกล้เคียงเฟรมมากที่สุด ตามที่กำหนดโดยอัตราเฟรมของกรุป ตัวอย่างเช่น สมมุติว่าวิดีโอกรุปมีอัตราเฟรม 30 fps แต่ละเฟรมประมาณ 33 มิลลิวินาที (ms) ถ้าเราเพิ่มวิดีโอ 1.68 วินาทีเข้าไปใน Timeline เริ่มต้นที่เวลา 0 source ไม่ได้จบตรงเฟรมพอดี DES กะประมาณเวลาหยุดที่ 1.6666 วินาที (50 เฟรม) ถ้าเราหาเวลาที่ 1.68 วินาทีก็จะได้เวลาจบของ source ที่แท้จริงคือ เฟรมที่ 51

อย่างไรก็ตาม DES จะไม่ overwrite เวลาหยุดของ source เราอาจจะเปลี่ยนอัตราเฟรมของกรุปหรือย้าย source ไปยังจุดใหม่ใน Timeline ที่ซึ่งจะมีการกะประมาณที่ต่างไป ด้วยเหตุนี้ DES จะสงวนเวลาหยุดและกะประมาณเท่าที่จำเป็นเท่านั้น

3.2 ออบเจกต์ที่ใช้งานตัดต่อวิดีโอของ DES

DES สนับสนุนออบเจกต์ต่อไปนี้สำหรับการตัดต่อวิดีโอ

- AMTimeline
- AMTimelineComp
- AMTimelineEffect
- AMTimelineGroup
- AMTimelineObj
- AMTimelineSrc
- AMTimelineTrack
- AMTimelineTrans

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- IAMTimelineEffectable
- IAMTimelineSplittable
- IAMTimelineTransable
- IAMTimelineVirtualTrack
- MediaDet
- MediaLocator
- RenderEngine
- Xml2Dex

บางออบเจ็กต์ต้องอ้างอิงกับ "Dexter 1.0 Type Library" (Qedit.dll)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบโปรแกรม

4.1 หลักการและแนวคิดการออกแบบ

ในการทำโครงการนี้ผู้จัดทำได้เลือกใช้ภาษาวิซวลเบสิก โดยสามารถใช้ได้กับไคเรกโซว์ เอพีไอ อีกทั้งมีโครงสร้างที่ไม่ซับซ้อน การเขียนโปรแกรมกับไคเรกโซว์ ด้วยภาษาวิซวลเบสิกนั้นต้องทำความเข้าใจกับ COM ออบเจกต์และอินเทอร์เฟซต่างๆในการที่เราจะดำเนินการกับสตรีมข้อมูลนั้น ซึ่งจะกระทำผ่านเมธอดของออบเจกต์ที่ได้ประกาศให้เป็นอินสแตนซ์ (instance) ของอินเทอร์เฟซ

หลักการออกแบบเมื่อพิจารณาโปรแกรมจากการใช้งานแล้ว ผู้ใช้โปรแกรมสามารถใช้งานได้ดังนี้

การใช้งาน	คำอธิบาย
เล่นวิดีโอ (play)	เริ่มเล่นวิดีโอจากตำแหน่งที่ต้องการ
หยุดเล่นวิดีโอ (stop)	หยุดเล่น
หยุดวิดีโอค้าง (pause)	หยุดค้างวิดีโอที่ตำแหน่งในขณะนั้น
เก็บภาพ	เก็บภาพนิ่งของวิดีโอที่กำลังแสดงอยู่ในขณะนั้น
ตัดวิดีโอ	สามารถตัดวิดีโอโดยกำหนดจุดเริ่มและจุดสิ้นสุดที่ต้องการ

ตารางที่ 4-1 แสดงการทำงานของโปรแกรม

4.2 การทำงานของอินเทอร์เฟซและออบเจกต์ที่สำคัญ

4.2.1 ออบเจกต์ MediaDet

ทำหน้าที่เรียกหรือดึงข้อมูลเกี่ยวกับไฟล์มีเดีย เช่น จำนวนของสตรีม ระยะเวลา และอัตราเฟรมของแต่ละสตรีม การใช้ออบเจกต์นี้ต้องใช้ Qedit.dll และออบเจกต์นี้ยังมีเมธอดสำหรับดึงเฟรมจากวิดีโอสตรีมด้วย คุณสมบัติและเมธอดที่ใช้มีดังนี้

1. คุณสมบัติ Filename เพื่อแทนชื่อของไฟล์ที่ใช้โดยออบเจกต์ MediaDet
2. คุณสมบัติ StreamLength แทนระยะเวลาของสตรีม มีหน่วยเป็นวินาที
3. คุณสมบัติ Framerate เป็นอัตราเฟรมของสตรีม มีหน่วยเป็นเฟรมต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมธอด WriteBitmapBits ค้างเฟรม ณ เวลาใดๆในสตรีมและบันทึกเป็นไฟล์ เมธอดนี้ใช้ในการเก็บภาพ

4.2.2 ออบเจ็กต์ IBasicAudio

สนับสนุนส่วนประกอบด้านเสียงของฟิลเตอร์ กราฟทำให้สามารถปรับแต่งค่าความดังและความสมดุล ในการใช้ออบเจ็กต์นี้ต้องประกาศตัวแปรชนิด IBasicAudio และกำหนดให้เท่ากับออบเจ็กต์ FilgraphManager ออบเจ็กต์นี้ต้องอ้างอิง Quartz.dll คุณสมบัติที่ใช้มีดังนี้

1. Balance ตั้งค่าหรือเรียกดูค่าความสมดุลเสียง
2. Volume ตั้งค่าหรือเรียกดูค่าความดังเสียง

4.2.3 ออบเจ็กต์ IBasicVideo

สนับสนุนส่วนประกอบด้านเสียงของฟิลเตอร์ กราฟในการใช้ออบเจ็กต์นี้ต้องประกาศตัวแปรชนิด IBasicVideo และกำหนดให้เท่ากับออบเจ็กต์ FilgraphManager ออบเจ็กต์นี้ต้องอ้างอิง Quartz.dll คุณสมบัติที่ใช้มีดังนี้

1. VideoHeight เรียกดูข้อมูลขนาดความสูงของวิดีโอ
2. VideoWidth เรียกดูข้อมูลขนาดความกว้างของวิดีโอ

4.2.4 ออบเจ็กต์ IMediaEvent

ประกอบด้วยเมธอดสำหรับเรียกดูเหตุการณ์ที่เกิดขึ้นและสำหรับจัดการกับสิ่งที่เกิดขึ้นกับ Filter Graph Manager แอปพลิเคชันสามารถใช้ออบเจ็กต์นี้สำหรับตอบสนองกับเหตุการณ์ที่เกิดขึ้นในฟิลเตอร์ กราฟ เช่น เมื่อจบสตรีม หรือเกิดความผิดพลาดจากการเรนเดอร์กราฟ ในการใช้ออบเจ็กต์นี้ต้องประกาศตัวแปรชนิด IMediaEvent และกำหนดให้เท่ากับออบเจ็กต์ FilgraphManager ออบเจ็กต์นี้ต้องอ้างอิง Quartz.dll เมธอดที่ใช้คือ WaitForCompletion ซึ่งจะคอยจนกระทั่งฟิลเตอร์ กราฟพร้อมที่จะเรนเดอร์ข้อมูลทั้งหมด

4.2.5 ออบเจ็กต์ IVideoWindow

จัดการกับหน้าต่างที่ใช้สำหรับเรนเดอร์วิดีโอกราฟ ในการใช้ออบเจ็กต์นี้ต้องประกาศตัวแปรชนิด IVideoWindow และกำหนดให้เท่ากับออบเจ็กต์ FilgraphManager ออบเจ็กต์นี้ต้องอ้างอิง Quartz.dll คุณสมบัติที่ใช้มีดังนี้

1. Height ตั้งค่าหรือเรียกดูความสูงของหน้าต่างวิดีโอ
2. Left ตั้งค่าหรือเรียกดูพิกัดตามแกน x ของหน้าต่างวิดีโอ
3. Owner ตั้งค่าหรือเรียกดู parent window สำหรับหน้าต่างวิดีโอ
4. Top ตั้งค่าหรือเรียกดูพิกัดตามแกน y ของหน้าต่างวิดีโอ
5. Width ตั้งค่าหรือเรียกดูความกว้างของหน้าต่างวิดีโอ

6. WindowStyle ตั้งค่าหรือเรียกดูรูปแบบหน้าต่างของหน้าต่างวิดีโอ

4.2.6 อินเทอร์เฟซ IMediaControl

มีเมธอดที่ใช้สำหรับควบคุมการไหลของข้อมูลผ่านฟิลเตอร์ กราฟ รวมถึงเมธอดที่ใช้สำหรับเล่น, หยุดค้าง, หยุด กราฟ Filter Graph Manager จะสนับสนุนอินเทอร์เฟซนี้ เมธอดที่ใช้มีดังนี้

1. Run รันทุกฟิลเตอร์ในฟิลเตอร์ กราฟ
2. Pause หยุดค้างทุกฟิลเตอร์ในฟิลเตอร์ กราฟ
3. Stop หยุดทุกฟิลเตอร์ในฟิลเตอร์ กราฟ
4. RenderFile สร้างฟิลเตอร์ กราฟเพื่อเรนเดอร์ไฟล์ที่ระบุไว้

4.2.7 ออบเจกต์ IMediaPosition

ประกอบด้วยเมธอดที่ใช้สำหรับค้นหาในฟิลเตอร์ กราฟและตั้งค่าอัตรา Playback ในการใช้ออบเจกต์นี้ต้องประกาศตัวแปรชนิด IMediaPosition และกำหนดให้เท่ากับออบเจกต์ FilgraphManager ออบเจกต์นี้ต้องอ้างอิง Quartz.dll คุณสมบัติที่ใช้มีดังนี้

1. CurrentPosition ตั้งค่าหรือเรียกดูตำแหน่ง playback
2. Duration เรียกดูระยะเวลาของสตรีม

4.2.8 ออบเจกต์ FilgraphManager

สร้างและควบคุมฟิลเตอร์ กราฟ โดยแอปพลิเคชันสามารถใช้งานเพื่อสร้างกราฟสำหรับเล่นไฟล์ เช่น เล่น, หยุดค้าง, หยุด หรือค้นหาฟิลเตอร์ที่มีอยู่ ออบเจกต์นี้ต้องอ้างอิง Quartz.dll เมธอดที่ใช้มีดังนี้

1. Run ทำให้ทุกฟิลเตอร์ในฟิลเตอร์ กราฟทำงาน
2. Pause หยุดค้างทุกฟิลเตอร์ในฟิลเตอร์ กราฟ
3. Stop หยุดทุกฟิลเตอร์ในฟิลเตอร์ กราฟ
4. RenderFile สร้างฟิลเตอร์ กราฟเพื่อเรนเดอร์ไฟล์ที่ระบุ

4.3 การทำงานของออบเจกต์ที่สำคัญที่ใช้ในการตัดต่อ

4.3.1 ออบเจกต์ AMTimeline

มีเมธอดสำหรับดำเนินการกับ Timeline ซึ่งเป็นออบเจกต์หลักของ DES โดย Timeline นั้นเป็นชุดของลำดับเวลา เช่น คลิปวิดีโอ คลิปเสียง เอฟเฟกต์ และ Transition ระหว่างคลิป Render engine ใช้ Timeline เพื่อสร้างฟิลเตอร์ กราฟ ซึ่งทำให้แอปพลิเคชันสามารถเรนเดอร์เอาต์พุตได้ ออบเจกต์นี้ต้องอ้างอิง Qedit.dll เมธอดที่ใช้มีดังนี้

1. AddGroup เพิ่มกรุปใน timeline
2. CreateEmptyNode สร้างออบเจกต์ใหม่ของ timeline

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ออบเจกต์ AMTimelineObj

มีเมธอดสำหรับดำเนินการกับออบเจกต์ของ Timeline ใน DES ทุกออบเจกต์ของ Timeline สนับสนุน AMTimelineObj รวมถึงพวกออบเจกต์ source เอฟเฟกต์ Transition แทร็ก กรุป และ Composition เราสร้างออบเจกต์ของ Timeline โดยเรียกใช้เมธอด AMTimeline.CreateEmptyNode ออบเจกต์นี้ต้องอ้างอิง Qedit.dll ส่วนเมธอดที่ใช้คือ SetStartStop2 เพื่อตั้งค่าเวลาเริ่มต้นและเวลาหยุดของออบเจกต์ซึ่งเกี่ยวข้องกับ parent ของออบเจกต์

4.3.3 ออบเจกต์ AMTimelineSrc

มีเมธอดสำหรับจัดการและตั้งค่าคุณสมบัติของ source ซึ่งออบเจกต์ AMTimelineSrc จะแทนคลิปที่มาจาก 1 สตรีมในมีเดีย เราสามารถใช้ส่วนของข้อมูลภายในไฟล์ source โดยการตั้งค่าเริ่มต้นและสิ้นสุดของ Media time ค่าเหล่านี้จะระบุเวลาเริ่มต้นและสิ้นสุดของออบเจกต์ AMTimelineSrc ซึ่งเกี่ยวข้องกับมีเดีย Media time อาจแตกต่างจากเวลาเริ่มและสิ้นสุดใน Timeline ของออบเจกต์ โดยยอมให้การเคลื่อนไหวเร็วหรือช้าได้ (ถ้าเป็นเสียงระดับเสียงจะมีการเลื่อน) ออบเจกต์นี้ต้องอ้างอิง Qedit.dll เมธอดที่ใช้มีดังนี้

1. SetMediaName ระบุชื่อของไฟล์ source
2. SetMediaTimes2 ตั้งค่าเวลาเริ่มต้นและสิ้นสุดของมีเดีย

4.3.4 ออบเจกต์ AMTimelineTrack

มีเมธอดสำหรับจัดการกับแทร็ก แทร็กนั้นจะประกอบด้วยรายการของ source ที่จะถูกเรนเดอร์ออกมาเป็นเอาต์พุต source ที่อยู่ภายในแทร็กเดียวกันอาจจะไม่ซ้อนกัน แทร็กวิดีโอสามารถมีทั้งเอฟเฟกต์และ Transition Render engine จะใช้เอฟเฟกต์ก่อนที่จะใช้ Transition แทร็กเสียงอาจจะมีเอฟเฟกต์แต่ไม่มี Transition เมธอดที่ใช้คือ SrcAdd จะทำการเพิ่ม source เข้าไปในแทร็ก

4.3.5 ออบเจกต์ AMTimelineGroup

ตั้งค่าและเรียกคุณสมบัติของกรุป โดยจะประกอบด้วย 1 แทร็กหรือมากกว่าและอาจจะมี 1 Composition หรือมากกว่าซึ่งจะประกอบด้วยคลิปชนิดเดียวกัน เช่น วิดีโอหรือเสียง กรุปเป็นส่วนประกอบที่อยู่ส่วนบนสุดใน Timeline และยังสามารถสนับสนุนออบเจกต์ AMTimelineComp timeline สามารถมีได้หลายกรุป ออบเจกต์นี้ต้องอ้างอิง Qedit.dll เมธอดที่ใช้มีดังนี้คือ

1. SetMediaTypeForVB ระบุชนิดของมีเดียในกรุป
2. SetRecomFormatFromSource ตั้งค่ารูปแบบการบีบอัดวิดีโอโดยใช้รูปแบบเดียวกับไฟล์ source

4.3.6 ออบเจกต์ AMTimelineComp

เพิ่มหรือเรียกดู Virtual track ใน Composition โดย Composition เป็นชุดของเลเยอร์ที่ประกอบด้วยหลายแทร็ก ยกตัวอย่างเช่น Composition ที่ประกอบด้วย 2 แทร็กกับ Transition ระหว่างแทร็กจะเปรียบเสมือนเป็นแทร็กเดี่ยวพร้อมด้วย Transition ก่อนที่จะรวมกัน Composition ควรจะประกอบด้วยมีเดียเพียงแค่นิดเดียว (วิดีโอหรือเสียง) แต่ข้อจำกัดนี้ไม่บังคับ Virtual track จะเป็นออบเจกต์ที่สามารถอยู่ภายใน Composition รวมทั้งแทร็กและ Composition อื่นๆได้

โหนดที่อยู่บนสุดใน Timeline คือ กรุป ซึ่งสนับสนุน AMTimelineComp และ AMTimelineGroup เมธอดที่ใช้คือ VTrackInsBefore โดยจะเพิ่ม Virtual track ใน Composition โดยระบุลำดับความสำคัญ

4.3.7 ออบเจกต์ SmartRenderEngine

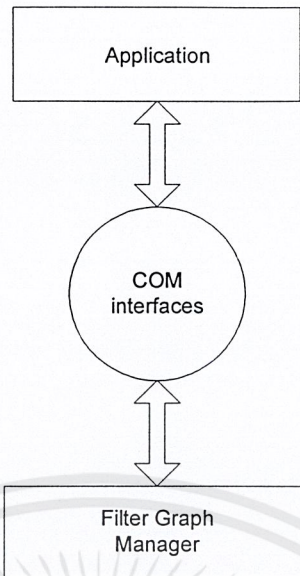
ออบเจกต์นี้จะทำหน้าที่สร้างเอาต์พุตที่ถูกบีบอัด โดยใช้ Smart recompression ซึ่งไฟล์ source จะถูกบีบอัดใหม่ถ้ารูปแบบแตกต่างจากรูปแบบของเอาต์พุต ถ้ามีรูปแบบตรงกันจะถูกบันทึกโดยตรงที่ไฟล์เอาต์พุต Smart recompression สามารถปรับปรุงเวลาในการเรนเดอร์โดยขึ้นอยู่กับสถานการณ์ ออบเจกต์นี้ต้องอ้างอิง Qedit.dll เมธอดที่ใช้คือ

1. ConnectFrontEnd ทำหน้าที่สร้างฟิลเตอร์ กราฟจาก timeline
2. GetFilterGraph จะเรียกดูฟิลเตอร์ กราฟที่ render engine สร้าง
3. SetTimelineObject จะตั้งค่า timeline เพื่อให้ render engine ใช้

ส่วนที่สำคัญของโปรแกรมมีดังนี้

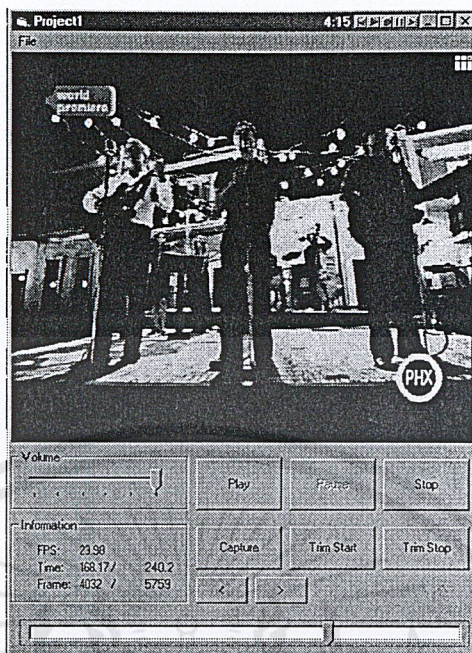
- ส่วนของการเรนเดอร์วิดีโอสามารถทำให้โดยสร้าง ออบเจกต์ FilgraphManager ขึ้นมาแล้วเรียกใช้ผ่านเมธอดหลัก play, pause ,stop
 - ส่วนของการควบคุมเสียง ใช้อินเทอร์เฟซ IBasicAudio เพื่อควบคุม ความดังและความสมดุล
 - ส่วนของการปรับสเกลและการเปลี่ยนวิดีโอเอาต์พุต สามารถกระทำโดยผ่านเมธอด IVideoWindow เพื่อปรับเปลี่ยนคุณสมบัติของหน้าต่างแสดงผล
 - การติดตามตำแหน่งของวิดีโอใช้อินเทอร์เฟซ IMediaPosition เรียกใช้เมธอด CurrentPosition เพื่อให้ผู้ใช้ปรับจุดที่ภาพจะเริ่มแสดง ใช้เมธอด Rate เพื่อให้ผู้ใช้สามารถปรับอัตราเล่นที่ภาพถูกแสดง
 - การดึงภาพเก็บไว้ต้องสร้างออบเจกต์ MediaDet ซึ่งเก็บข้อมูลเกี่ยวกับมีเดียไฟล์และเก็บภาพผ่านเมธอด WriteBitmapsBits โดยจะเป็นภาพประเภทบิตแมป
 - การตัดต่อวิดีโอจะใช้ออบเจกต์ที่อยู่ใน DES เช่น AMTimeline, AMTimelineobj
- แสดงรูปการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



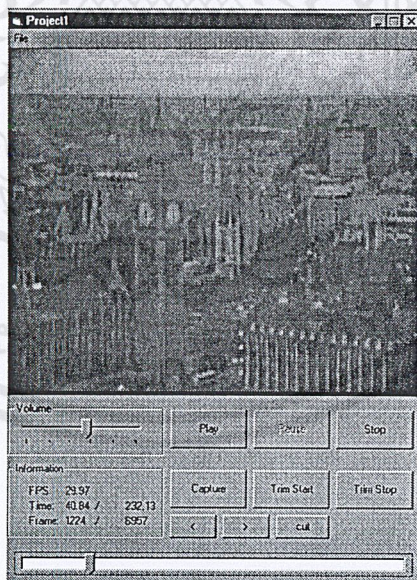
รูปที่ 4-1 การติดต่อกับฟิลเตอร์ กราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-2 เมื่อ Playback วิดีโอ

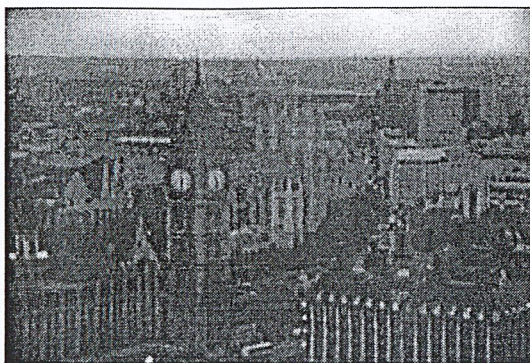
4. ทดลองเก็บภาพโดยทำการเลื่อนสไลเดอร์เพื่อค้นหาภาพที่ต้องการ โดยหากเฟรมหรือเวลา



รูปที่ 5-3 แสดงการค้นหาภาพ

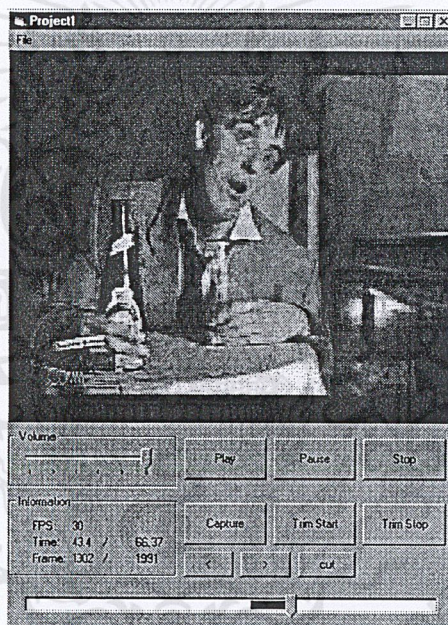
5. ทำการเก็บภาพโดยกดปุ่ม Capture จะได้ภาพที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-4 แสดงภาพที่ทำการ Capture

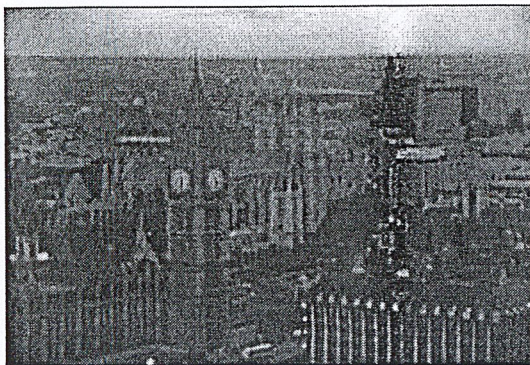
6. ทดลองตัดวิดีโอโดยกำหนดจุดที่ต้องการ โดยใช้ปุ่ม Trim Start และ Trim Stop เพื่อกำหนดจุดเริ่มต้นและจุดสิ้นสุด



รูปที่ 5-5 ภาพขณะเล่นวิดีโอ

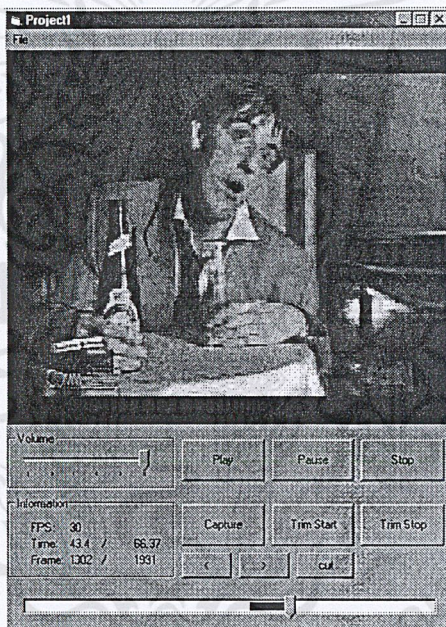
7. ทดลองเปิดไฟล์วิดีโอที่ตัดแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-4 แสดงภาพที่ทำการ capture

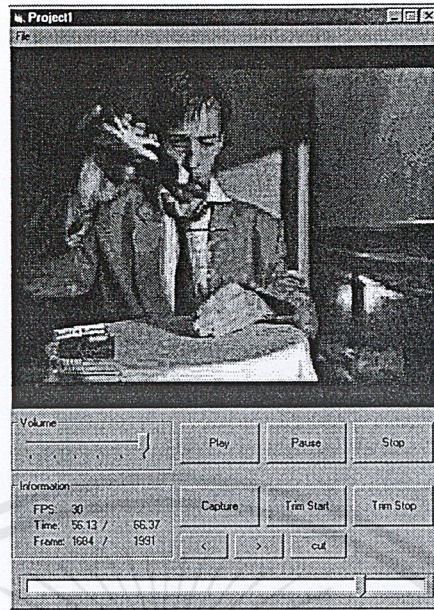
6. ทดลองตัดวิดีโอโดยกำหนดจุดที่ต้องการ โดยใช้ปุ่ม Trim Start และ Trim Stop เพื่อกำหนดจุดเริ่มต้นและจุดสิ้นสุด



รูปที่ 5-5 ภาพขณะเล่นวิดีโอ

7. ทดลองเปิดไฟล์วิดีโอที่ตัดแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-6 แสดงวิดีโอที่ตัดออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปและวิจารณ์

6.1 บทวิจารณ์

จากโปรแกรมที่ได้พัฒนามาเพื่อใช้สำหรับตัดต่อวิดีโอ ผลที่ได้ออกมาโปรแกรมสามารถที่จะ Playback วิดีโอและสามารถค้นหาภาพจากเฟรมหรือเวลาที่ต้องการเพื่อเล่นหรือเก็บภาพที่ต้องการได้ โดยภาพที่ดึงมานั้นจะมีขนาดเท่ากับขนาดจริงของภาพนั้น ในการตัดวิดีโอสามารถทำการตัดได้ แต่มีข้อจำกัดอยู่ว่าตัดได้เพียงแค่ช่วงสั้นๆเท่านั้น ซึ่งมีสาเหตุจากการที่เมธอดของออบเจกต์ที่ใช้ตัดต่อไม่สนับสนุนรูปแบบของวิดีโอที่ใช้ตัดต่อ ทำให้บางวิดีโอที่ตัดออกมาแล้วไม่สามารถเปิดดูได้และยังมีขนาดไฟล์ที่ใหญ่มาก และข้อจำกัดอีกอย่างหนึ่งคือเมื่อตัดวิดีโอที่มีขนาดความยาวของช่วงที่เลือกหลายๆ จะใช้เวลานานมากในการทำงาน โปรแกรมจึงเหมาะกับการตัดวิดีโอขนาดไม่ใหญ่จนเกินไป

6.2 แนวทางในการพัฒนาโปรแกรม

แนวทางในการพัฒนาต่อนั้นควรจะหาวิธีทำให้เรนเดอร์ภาพขนาดเท่าภาพจริงได้จะทำให้สามารถชมภาพในขนาดจริงได้ การตัดวิดีโอต้องปรับปรุงให้วิดีโอที่ตัดออกมามีขนาดไฟล์ไม่ใหญ่จนเกินไปและตัดให้ได้เวลามากขึ้นตลอดจนสามารถตัดวิดีโอที่มีขนาดความยาวของช่วงที่เลือกหลายๆได้ และอาจพัฒนาในแง่ของการใช้งานให้สะดวกยิ่งขึ้นเนื่องจากโปรแกรมจะสามารถตัดต่อได้ที่ละไฟล์เท่านั้น หรือหาวิธีการตัดต่อภาพที่ไม่ซับซ้อน โดยใช้เทคนิคหรือเขียนด้วยภาษาอื่นๆ

6.3 บทสรุป

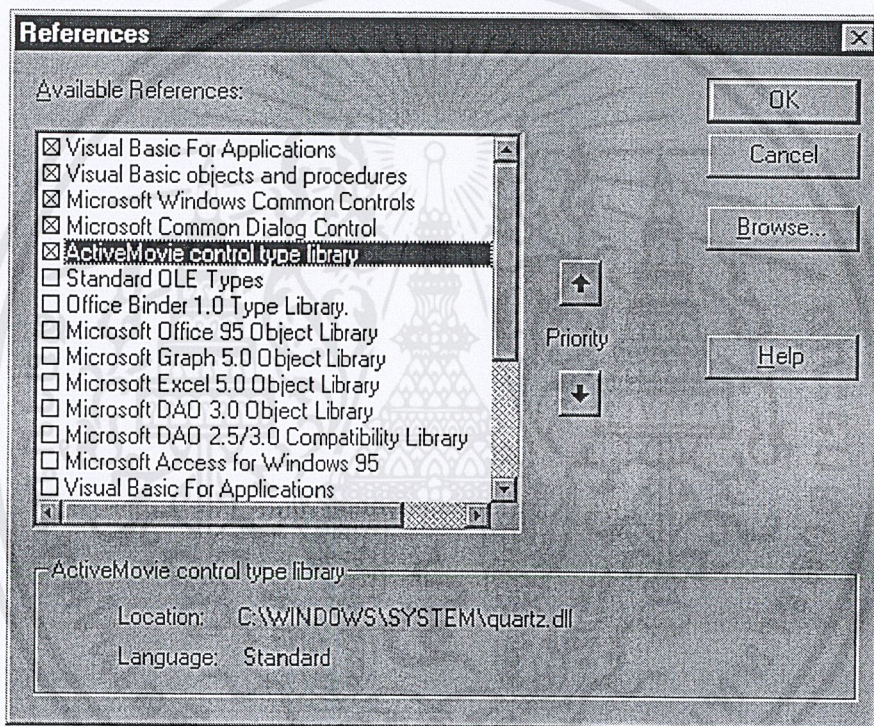
โปรแกรมตัดต่อวิดีโอสามารถตอบสนองความต้องการส่วนบุคคลได้อย่างเพียงพอ และเหมาะสำหรับผู้ต้องการทำคลิปวิดีโอไว้ชมส่วนตัวเพราะใช้งานง่าย สามารถทำเองที่บ้านได้ แต่ถ้ามีการพัฒนาต่อไปโดยใช้เทคนิคอื่นๆเข้ามาช่วยก็จะมีประสิทธิภาพเหมือนกับซอฟต์แวร์ที่ขายตามท้องตลาด และผลงานที่ออกมาก็จะสวยงามกว่า การพัฒนาต่อสามารถพัฒนาโดยการถอดสัญญาณวิดีโอหรือใช้ชุด SDK ของใครก็ได้ที่มีเอพีไอที่เหมาะสมกับการดำเนินการกับสตรีมข้อมูลให้เลือกใช้

ภาคผนวก ก

การติดตั้ง Quartz.dll, Qedit.dll เพื่อใช้งานอินเทอร์เฟซของไคเรกโซว์

เพื่อที่จะใช้งานอินเทอร์เฟซของไคเรกโซว์ในวิชวลเบสิก แอปพลิเคชัน เราจะต้องติดตั้ง Microsoft ActiveMovie control type library ในวิชวลเบสิก โดยการเลือกเมนู Project > References และเลือกที่ ActiveMovie control type library

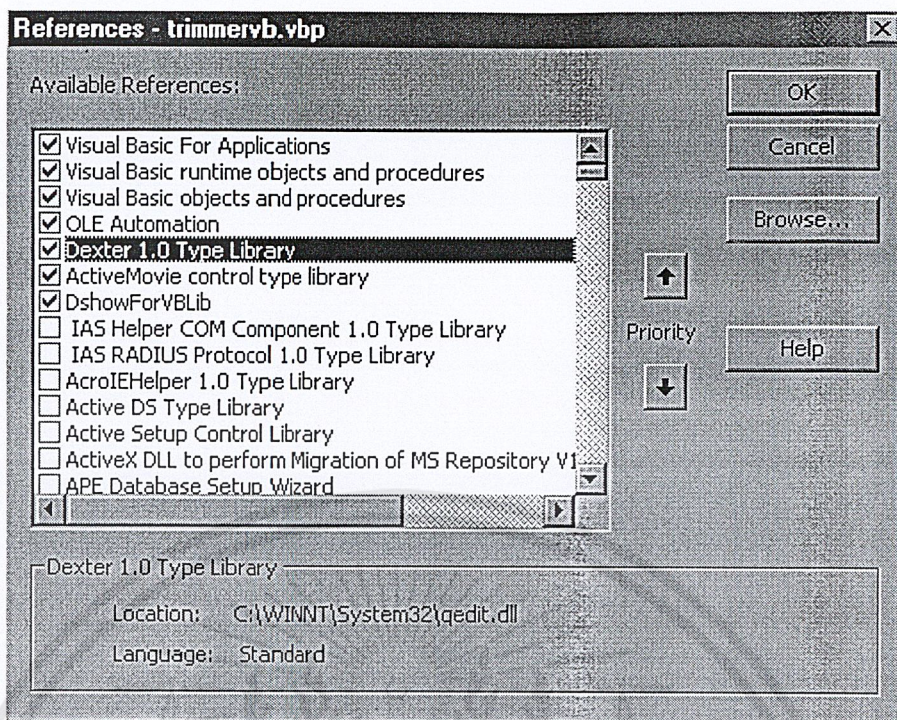
เมื่อติดตั้งโดยเลือกไอคอน บ็อกซ์ References นั้นหมายความว่าเรากำลังระบุไลบรารีที่มีข้อมูลที่วิชวลเบสิกต้องการ โดยอัตโนมัติ รูปต่อไปนี้แสดงไอคอน บ็อกซ์ References



รูปแสดง การติดตั้ง Quartz.dll

หลังจากติดตั้งไลบรารีเสร็จเรียบร้อยแล้ว สามารถที่จะใช้ไอคอน บ็อกซ์ Object Browser เพื่อที่จะดูเมธอด, อีเวนต์และคุณสมบัติที่ใช้กับอินเทอร์เฟซที่เกี่ยวข้องได้ เลือก Object Browser จากเมนู View จากนั้นเลือก QuartzTypeLib จากรายชื่อไลบรารีที่มีอยู่ที่ด้านบนซ้ายของไอคอน Object Browser โปรดสังเกตว่าข้อมูลที่อยู่ใน QuartzTypeLib รวบรวมตามอินเทอร์เฟซมากกว่าออบเจกต์

ส่วนการติดตั้ง Qedit.dll นั้นก็ทำแบบเดียวกัน โดยหลังจากเลือก References แล้วให้เลือกที่ Dexter 1.0 Type Library เพื่อให้สามารถใช้งานเมธอด, อีเวนต์หรือคุณสมบัติต่างๆของอินเทอร์เฟซที่เกี่ยวข้องกับงานตัดต่อได้



รูปแสดง การติดตั้ง Qedit.dll

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Joan L. Mitchell, William B. Pennebaker, Chad E. Fogg, Didier J.LeGall : “*MPEG Video Compression standard*” , International Thomson Publishing.
- [2] Microsoft Developer Network Library (MSDN) , January 2000
- [3] Microsoft DirectX 8.1 SDK Help Document
- [4] จัฑทวุฒิ พิษผล, พิชิต สันติภูพานนท์ : “คู่มือเรียน Visual Basic 6”, PROVISION, กรุงเทพมหานคร, มกราคม 2544

เว็บไซต์อ้างอิง

- [1] <http://www.microsoft.com>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้